a站项目文档

项目小组 xx小组

小组成员 宁志豪、何玉洁、刘苏云、代贝妮、吴华、陈海洪、叶涵予

联系方式

重庆师范大学软件工程系

摘要

面对综合视频平台内容泛娱乐化、垂直领域用户缺乏高质量互动与社区归属感的行业背景，本项目旨在为 16-28 岁 Z 世代的兴趣群体打造一个专注于二次元、科技等领域的垂直视频社区——A 站。项目目标是通过构建“高质量视频+实时弹幕+深度社群”的核心模式，解决内容消费者“找同好难”与创作者“分享渠道少”的双重痛点。

开发过程采用分阶段迭代策略：首阶段聚焦 PC 端，完成视频上传播放、弹幕互动、用户中心等最小可行产品；后续阶段逐步扩展社区功能与移动端。技术栈上，选用 C++/QML、JavaScript 与 MySQL 等成熟技术，确保开发效率与系统稳定性。

项目成果预期为一个具备独特社区文化、高用户粘性的兴趣平台，为内容生态与商业化奠定基础，具备较好的市场可行性与发展前景。

关键词：垂直视频社区，弹幕互动，Z 世代，兴趣社群，迭代开发

日期	修改	描述	作者
2025 年 9 月 5 号	0.1.0	初始版本，明确基本目标	
2025 年 10 月 16 号	0.1.2	进一步完善需求文档	
2025 年 10 月 19 号	0.2	对需求文档进行大改：愿景文档、需求金字塔、涉众分析、用况建模、基本流、备选流	
2025 年 10 月 30 日	0.2.1	更新文档结构	
2025 年 11 月 23 日	0.2.2	对用例进行大的修改	

目录

1 立项

1.1. 项目起源与提案

1.1.1 解决需求与痛点

外部需求：市场上综合视频平台内容泛娱乐化，垂直领域（二次元、硬核科技等）爱好者和深度创作者缺乏高质量互动环境与同好圈子，存在“社区归属感弱”的痛点。

用户痛点：内容消费者难以获得垂直领域高质量内容与实时互动体验，内容创作者缺少便捷分享渠道与公平流量机制。

1.1.2 捕捉机会

市场驱动：互联网视频内容消费持续增长，Z 世代及年轻用户（16-28 岁）对“深度互动+兴趣社区”需求旺盛，存在未被满足的细分市场空白。

战略驱动：布局视频社区赛道，通过“核心兴趣模块”构建差异化竞争力，为未来拓展直播、电商、游戏等商业化模块奠定基础。

1.1.3 商业价值驱动

增收：短期通过基础功能积累用户，长期可通过会员、直播打赏、创作者激励等模块直接创造收入，同时提升用户留存间接促进商业价值增长。

增效：通过自动化内容审核、云服务器存储与管理，替代部分人工操作，降低平台运营成本；借助协同工具提升团队开发与运营效率。

1.2. 项目提案

1.2.1 发起方

由我们七人团队发起。

1.2.2 提案形式

结合“business case”，明确项目愿景、目标、功能模块与实施路径，而非单一原型。

1.2.3 核心内容

1.2.3.1 解决的问题：

垂直领域用户“缺高质量内容、缺互动、缺归属感”，创作者“缺工具、缺流量、缺反馈”的双重痛点。

1.2.3.2 目标用户

核心为 16-28 岁学生与年轻上班族（内容消费者）、平台生态基石的内容创作者（UP 主）。

1.2.3.3 解决方案

打造以“兴趣模块”为核心的视频社区，核心功能涵盖视频上传/播放/弹幕互动、创作者后台、用户中心，分三阶段落地（首阶段聚焦电脑端核心功能，后续拓展手机端）。

1.2.3.4 时机

当前视频内容消费增长，细分市场需求未被满足，先发布局电脑端可快速验证需求，抢占“垂直社区”赛道先机。

1.2.3.5 预期商业价值

短期积累核心用户与创作者，中期通过会员、直播等实现增收，长期构建“内容+社区+商业化”生态，提升市场份额。

1.2.3.6 初步资源评估

技术栈确定为后端 C++/QML/JavaScript、前端 QML、云数据库 MySQL、云服务器、云存储；人力为 7 人团队，分阶段投入开发，首阶段聚焦最小可行产品（电脑端核心功能）。

1.3. 项目可行性分析

1.3.1 市场可行性

目标用户（Z 世代）对弹幕互动、兴趣社区接受度高，且细分市场竞争压力较小，需求真实存在；电脑端用户基数稳定，先开发电脑端可覆盖核心办公、居家使用场景。

1.3.2 选定的技术栈

（C++、MySQL、Socket 等）成熟，电脑端核心功能（视频播放、弹幕、基础审核）无技术壁垒，可依托云服务器实现快速部署。

1.3.3 经济可行性

首阶段仅开发电脑端，暂不涉及手机端与高级算法，成本可控；长期可通过会员、直播等模块实现盈利，ROI（投资回报率）具备增长潜力。

1.3.4 运营可行性

审核员、管理员角色分工明确，基础审核流程与用户运营机制清晰，可支撑电脑端初期平台运转。

1.4. 项目 Business Case

1.4.1 核心定位

1.4.2 关键组成部分

1.4.2.1 业务建模

现状分析：用流程图呈现当前“垂直领域用户分散、创作者分享难”的痛点，例如“用户找垂直内容→浏览泛娱乐平台→无法精准匹配”的低效流程，可视化痛点所在；同时体现“电脑端仍是部分用户（如学生、办公族）观看长视频、创作内容的主要场景，却缺乏专属垂直社区”的现状缺口。

未来设计：通过新业务模型图展示“用户（电脑端）→兴趣频道→精准内容→弹幕互动→关注创作者”的高效流程，以及“创作者（电脑端）→便捷上传→审核→流量分发→粉丝互动”的闭环，直观呈现解决方案。

工具应用：基于“商业模式画布”9 大模块构建逻辑，例如“客户细分”为 16-28 岁垂直领域用户与创作者，“价值主张”为“电脑端优质体验+弹幕互动+兴趣社区+创作者友好工具”，“收入来源”短期为会员，长期拓展直播打赏等。

1.4.2.2 详细财务分析

成本：首阶段（电脑端开发）云服务器租赁、人力开发成本；中期（手机端迭代）额外开发与适配成本；长期（功能拓展）运营与维护成本。

收益：量化会员付费、直播打赏等收入预期，计算ROI与投资回收期，例如“预计电脑端上线1年用户达x万，会员转化率x%，单会员年费x元，年会员收入x万；手机端上线后用户规模预计提升x%，收入同步增长”。

1.4.2.3 全面风险评估

风险类型：技术风险（弹幕屏蔽算法实现难度）、市场风险（电脑端用户留存不及预期）、合规风险（内容审核疏漏）、迭代风险（电脑端与手机端数据同步问题）。

缓解策略：技术风险通过“先基础屏蔽功能，后迭代算法”解决；市场风险通过“聚焦核心垂直领域+电脑端专属活动”提升粘性；合规风险通过“人工审核+关键词自动检测”规避；迭代风险通过“统一数据接口设计”确保多端数据互通。

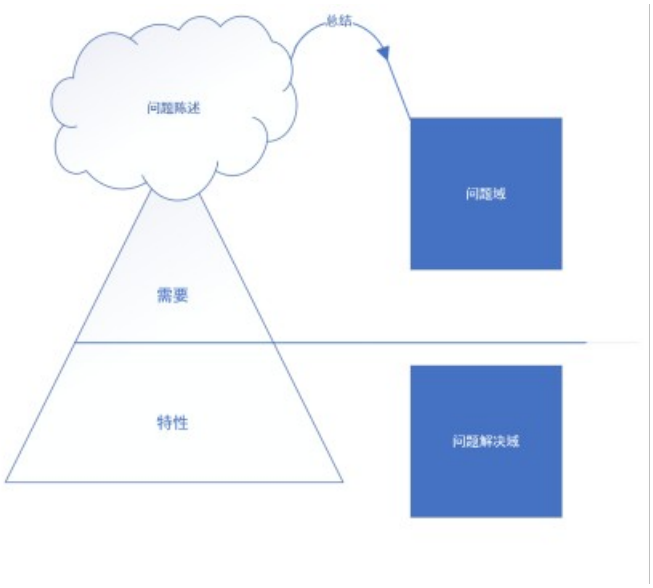
1.4.2.4 实施计划

时间线：首阶段（8个月-1年）完成电脑端核心功能（视频上传/播放、弹幕、基础审核、用户中心）；第二阶段上线电脑端社区系统（评论、关注、动态）；第三阶段开发手机端，同步推进付费与直播功能。

资源需求：首阶段需7人团队（开发、测试、运营），云服务器（腾讯云），MySQL云数据库，云存储等；手机端开发阶段需额外补充移动端开发人力。

2 愿景

2.1. 问题陈述与需求金字塔



2.1.1 问题一

要素	描述
问题	内容消费者缺乏能进行高质量欣赏和实时互动、方便联系同好的平台。
影响	难以在社区中找到契合自身兴趣的高质量内容与互动场景，无法充分满足文化消费需求，对社区的参与感和融入感较低。
结果	可能逐渐减少在该社区的停留时间，甚至放弃使用该社区，转而寻找其他能更好满足需求的平台，导致社区用户流失。
优点	用户能在社区中获得丰富、优质且符合兴趣的内容，通过实时互动与同好深入交流，增强对社区的认同感和归属感，提高用户粘性和活跃度，促进社区文化的繁荣发展。

2.1.2 问题二

要素	描述
问题	内容创造者缺少社区对应的分享平台来分享个人见解或展示自己、寻找同好的渠道。
影响	创作内容缺乏有效的展示和传播途径，难以吸引目标受众，创作动力可能受挫，且难以与同领域创作者交流合作，限制创作水平的提升。
结果	部分创作者可能因缺乏发展机会而减少创作频率或放弃在该社区创作，导致社区优质内容供给不足，内容多样性受限，影响社区的整体吸引力。

优点	创作者拥有专属且高效的分享平台，能够精准触达目标受众，获得更多反馈和支持，激发创作热情，提升创作质量。同时，便于与同好交流，促进创作思路的拓展和创新，为社区带来丰富多样的优质内容。

2.1.3 问题三

要素	描述
问题	平台审核员面对海量用户生成内容，缺乏高效、精准且标准统一的工具与方法来确保内容安全与合规。
影响	审核效率低下，导致违规内容不能及时处理，影响社区内容质量；尺度把握不一，可能造成对合规内容的误判或对违规内容的放纵，引发用户不满，损害社区公平性和公信力。
结果	审核员工作压力大，可能出现职业倦怠，导致审核团队不稳定；社区内容安全问题频发，影响用户体验，降低用户对社区的信任度，甚至可能面临监管风险。
优点	审核员借助高效、精准且标准统一的工具与方法，能够快速准确地处理海量内容，提高审核效率和质量，确保内容安全与合规。减轻工作压力，提升工作满意度和审核团队的稳定性，维护社区的良好秩序和公平环境，增强用户对社区的信任。

2.1.4 问题四

要素	描述
问题	系统管理员支撑高并发、强交互的视频社区平台时，缺乏对系统全局健康状况、用户权限和数据资源的集中、可视化管理能力。
影响	难以实时掌握系统运行状态，不能及时发现和解决系统故障，影响平台的稳定性和可用性；用户权限管理混乱，可能导致数据泄露、非法操作等安全问题；数据资源管理分散，不利于数据的整合和利用，影响平台的决策和发展。
结果	平台频繁出现故障，影响用户体验，导致用户流失；安全事件频发，损害平台声誉，面临法律风险和经济损失；数据资源无法有效利用，影响平台的业务创新和竞争力提升。
优点	系统管理员通过集中、可视化的管理界面，能够实时监控系统全局健康状况，及时发现并解决系统问题，确保平台的稳定运行。精准管理用户权限，保障数据安全，防止安全事件发生。有效整合和利用数据资源，为平台的决策和发展提供有力支持，提升平台的竞争力和可持续发展能力。

2.2. 涉众与用户

2.2.1 涉众

涉众	涉众类型	简要描述

视频消费与互动者（观看视频的人）	Users(用户们） ->主要服务对象，系统的主要使用者	获取高质量、感兴趣的内容（如动画、游戏、知识等）；流畅的播放与弹幕互动体验；个性化的内容推荐；强烈的社区归属感
Up 主（内容创作者）		便捷的视频上传、管理和数据分析工具；公平的流量分发与变现机制（如创作激励）；与粉丝的有效互动渠道；良好的社区氛围以支持持续创作
审核员/内部运营人员/		高效的内容审核工具与明确的规范，以维护社区氛围与安全；丰富的后台数据以支持运营决策
投资人/股东/甲方（假定，实际上没有）	Sponsors(发起人）->我们七人	他们往往关注视频平台的用户增长情况，市场份额盈利情况和商业回报等内容
业务经理这类人		分析平台在我们视频行业的独特定位和优势，从而指导开发出更适应市场的平台
包括开发，测试，运维->我们七人	Developers(开发人员）	我们关注的是一个清晰，稳定的产品需求和技术方案；合理的愿景开发周期；以及可维护可扩展的系统架构
内部监管机构（审核员）	Authorities(权威)	平台可以审核通关的内容应该符合法律法规，特别是版权和青少年保护什么的
领域专家（假定，实际上多半没有，对于社区文化的研究者）		他负责保证我们项目可以保持独特的社区文化模块（比如二次元社区，弹幕文化等），用于体现我们项目的核心价值
消费者（购买大会员或者推流的人，或者买直播打赏道具的人->暂定）	Customers(客户）	他负责保证我们项目可以保持独特的社区文化模块（比如二次元社区，弹幕文化等），用于体现我们项目的核心价值

2.2.2 用户角色与画像

角色	核心特征与目标	核心需求

游客	未登录用户，内容浏览者	1. 无需登录即可浏览首页和观看视频。 2. 能了解平台的基本内容和调性。 3. 被吸引后能方便地注册
普通用户	视频消费与互动者	1. 流畅地搜索和发现感兴趣的视频。 2. 通过弹幕、评论、点赞与他人互动。 3. 关注 UP 主，形成自己的信息流。 4. 管理自己的收藏夹和观看历史
UP 主	内容创作者，平台生态核心，已注册用户的泛化	1. 便捷地上传、管理和发布视频。 2. 获得清晰的数据反馈（播放量、粉丝数等）。 3. 与粉丝进行有效互动（评论、动态）
审核员	内部运营人员，内容安全守护者	1. 高效地审核用户上传的视频、评论、弹幕。 2. 对违规内容进行便捷处理（通过/驳回/封禁）。 3. 有明确的审核标准和操作指南
系统管理员	最高权限管理者	1. 管理所有用户和角色权限。 2. 配置系统全局参数。 3. 监控系统健康状况和数据报表

2.3. 关键涉众和用户的需要

2.3.1 视频消费者与互动者（如二次元爱好者）的需求

高质量内容获取：系统能够轻松发现并观看符合自身兴趣的高质量视频内容，满足娱乐、学习或深度文化消费的需求。

实时互动与归属感：在观看视频时，能通过弹幕、评论等方式与他人进行实时互动，感受“一起看”的陪伴感，并基于兴趣找到同好，建立强烈的社区归属感。

个性化体验：系统可以获得根据个人喜好定制的推荐内容，减少信息过载，提升内容发现的效率和愉悦感。

2.3.2 内容创作者（UP 主）的需要

便捷的创作与分享：系统拥有简单易用的工具来创作、上传和管理视频内容，并能轻松地将自己的见解或作品分享给目标受众，找到知音。

有效的反馈与成长：系统反馈清晰的播放量、粉丝增长、互动数据等反馈，以便让 up 主了解内容效果、优化创作方向，并建立个人影响力。

与粉丝深度互动：系统能够方便地让 up 主与粉丝进行交流（如回复评论、发布动态），维护和壮大自己的粉丝社群。

2.3.3 审核员的需要

高效与精准的审核：系统在面对海量内容时，能快速、准确地对视频、弹幕、评论进行合规性判断，显著提升审核效率，降低工作压力。

标准统一与决策支持：审核过程中有清晰、统一的规范和工具支持，确保审核尺度一致，减少误判和争议。

2.3.4 系统管理员的需要

系统稳定与可视化管理：系统能够全面监控平台运行状态（如服务器健康度、流量峰值），快速定位并处理问题，确保平台的高可用性和稳定性。

安全与可控的资源管理：系统具备强大的用户权限管理和数据资源控制能力，防止安全事件，保障平台和数据的安全有序。

2.4. 产品概述

2.4.1 产品定位陈述

for	**核心用户：**16-28 岁学生/年轻上班族（二次元、科技、知识领域爱好者） **细分群体：** 内容消费者（追求互动与归属感） UP 主（需要创作工具与流量支持） 审核员/系统管理员（保障内容安全与平台稳定）
who	**平台角色：** 兴趣社区连接者（弹幕+社群驱动） 创作生态赋能者（工具+数据支持） 内容安全守护者（审核+合规） **价值主张：** 对消费者：实时互动+个性化推荐 对创作者：公平流量+变现支持
the	**产品定位：** 以特定兴趣（如二次元、科技）为核心，集高质量视频、弹幕互动、深度社群于一体的垂直类视频社区平台，满足 Z 世代文化认同与深度互动需求。
That	**差异化优势：** 独特的弹幕文化（实时集体观看氛围） 强社区归属感（兴趣标签+社群） 创作者友好（简洁上传+数据反馈） 垂直领域聚焦（避免泛娱乐化竞争）

2.4.2 完整的产品概述

a 站是一款聚焦 Z 世代（16-28 岁）的垂直兴趣视频社区，以二次元、科技、知识等细分领域为核心，通过“高质量视频+实时弹幕+深度社群”模式，打造兼具娱乐性与文化归属感的兴趣聚集地。

目标用户涵盖内容消费者（追求互动与同好社群）、UP 主（需要创作工具与流量支持）及运营方（审核员、系统管理员）。消费者可通过智能推荐获取个性化内容，利用弹幕实时互动，并加入兴趣标签社群；UP 主可便捷上传视频、分析数据并获得创作激励；运营方则通过 AI+人工审核保障内容安全。

核心功能包括：

视频消费：多端观看、弹幕互动、智能推荐；

创作支持：一键上传、数据分析、收益分成；

社群运营：兴趣标签匹配、动态广场、私信互动；

管理保障：内容审核、青少年模式。

差异化优势在于：

弹幕文化：实时互动形成集体观看氛围，增强参与感；

社区属性：通过标签与社群构建深度连接，避免泛娱乐化；

创作者友好：简化流程、透明数据、公平流量，降低创作门槛；

垂直聚焦：深耕细分领域，精准满足用户需求。

发展愿景：短期积累 xxx 核心用户，签约 xxx 优质 UP 主；长期成为国内领先垂直兴趣社区，覆盖 xxx 用户，拓展电商、IP 衍生等增值服务，构建“内容-互动-变现”生态，实现可持续增长。

2.5. 产品特性 （需求**金字塔的底层**，是为了满足用户需要的功能）

对于内容消费者：可以提供智能推荐，比如经常点击哪个板块，还可以提供弹幕系统和聊天系统，增加内容消费者的参与程度，还可以添加点赞收藏评论功能，方便用户重复播放和分享个人观点或者吐槽，同时还可以关注 up 主，形成个人的动态视频流

对于 up 主（内容创作者）：可以提供视频上传工具，还可以分析视频的数据的工具进行反馈浏览，以及稿件的管理，封面与标题优化的建议，还有和粉丝互动的渠道，比如私信或者评论回复

对于审核员：应该有一个方便审核员审核的队列方便他驳回或者通过，并且可以添加驳回理由。或者可以的话实现一个内容详情的预览，方便审核是否违规或者插入自动检测屏蔽词自动驳回功能。

对于系统管理员：应该提供全局用户管理（封禁解封）以及系统的参数配置，云服务器情况，以及系统监控的仪表盘，数据的汇报情况导出或者展示功能

2.6. 其他产品需求

性能需求：

首页首屏加载时间 < 3 秒。

视频播放点击后开始缓冲时间 < 5 秒

安全性需求：

用户密码需进行处理后存储

敏感操作（如修改密码）需二次验证

可用性需求：

界面设计简洁直观，符合目标用户群体的审美

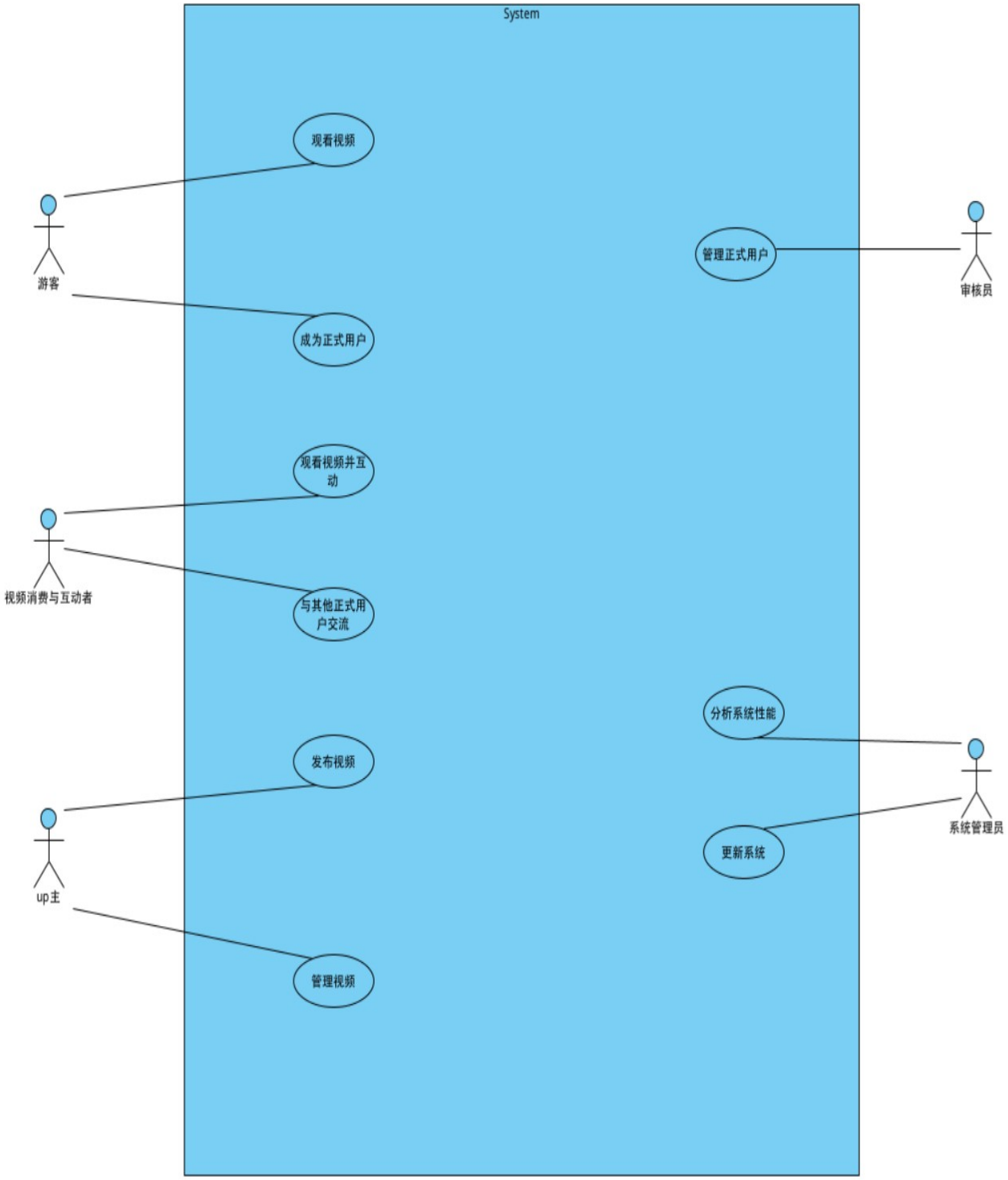
操作流程符合直觉，新用户无需培训即可完成核心操作（看视频、发弹幕）

3 用况建模

3.1. 术语表

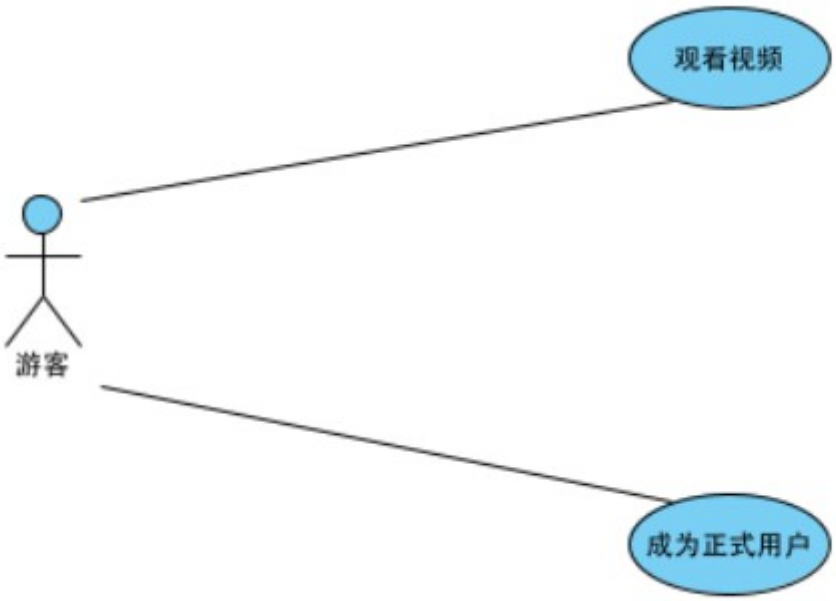
名称	定义
UP 主	平台的内容创作者
个人消息	包含所有的回应信息，系统通知，聊天信息
弹幕	在视频播放时实时飘过的评论
一键三连	对视频同时进行点赞、投币、收藏的操作
动态	UP 主发布的非视频类图文内容，类似微博
AV 号/BV 号	视频的唯一标识符
粉丝	普通用户关注 up 主之后会自动变为其粉丝
内容消费与互动者	区别于 up 主，此参与者主要是为了看视频和与视频互动
正式用户	包括 up 主和内容消费与互动者的更大群体

3.2. a 站的主要用况


System
观看视频
游客
成为正式用户
观看视频并互动
视频消费与互动者
与其他正式用户交流
发布视频
up主
管理视频
管理正式用户
审核员
分析系统性能
更新系统
系统管理员

3.3. 各部分用况图以及简要的用况和参与者描述

3.3.1 第一部分：**游客**

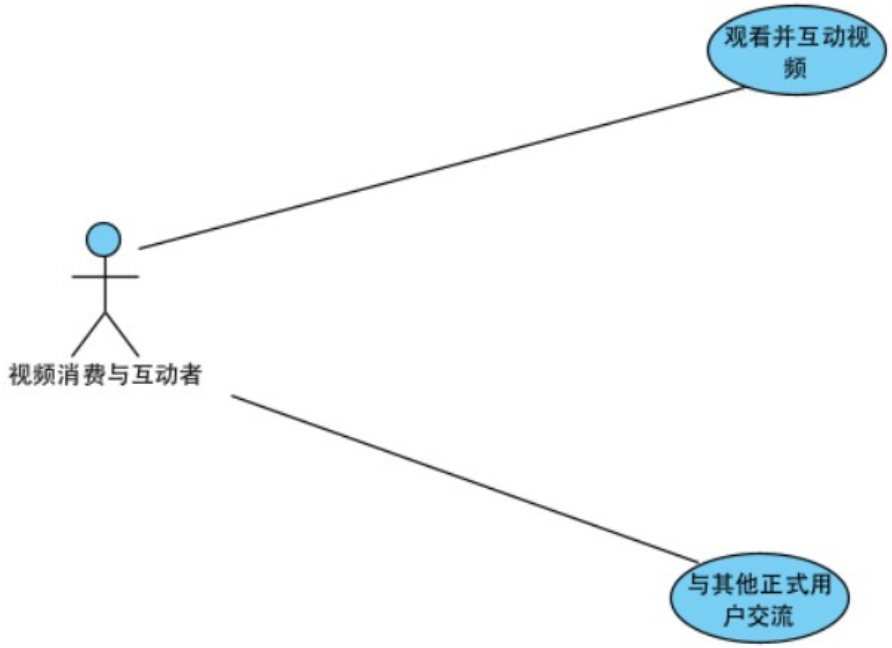


游客：游客是尚未经过注册，仅仅只是单纯为了观看视频的用户。

观看视频：游客可以通过搜索关键词查找公开视频，或者直接浏览并观看首页的公开视频。

成为正式用户：游客通过填写信息的方式，完成账户创建流程，从游客身份转变为**正式用户**，进而获取访问特定功能的权限。

3.3.2 第二部分：**视频消费与互动者**

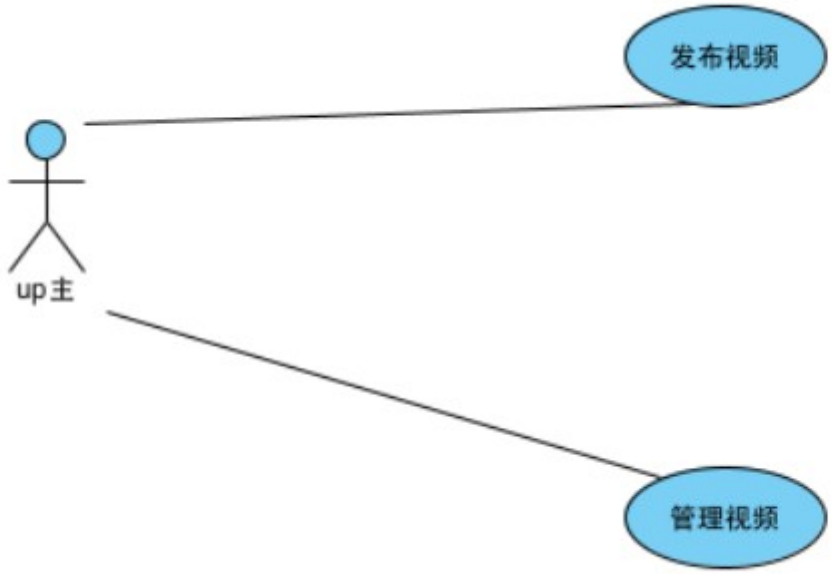


视频消费与互动者：视频消费与互动者是游客进行成为**正式用户**渠道后成为的，负责观看视频和与视频进行互动的人。

观看并互动视频：系统向视频消费与互动者提供了观看视频和与视频互动的功能，与视频互动包括了很多小功能，比如点赞，投币，收藏分享，关注和评论以及回复的功能。

与其他正式用户交流：系统在主页提供了动态和消息的按钮，视频消费与互动者可以发布动态和私信其他**正式用户**的功能。

3.3.3 第三部分：up 主



up 主（内容创作者）：up 主也是**正式用户**，是为了凸显 **up 主**和区别**视频消费与互动者**的参与者，他主要是分享视频到系统。

发布视频：系统向 **up 主**提供了可以上传视频的功能，系统验证视频格式并处理上传，视频进行审核成功后将推送到主界面。

管理视频：系统向 **up 主**提供了管理个人视频的内容的地方，可以对上传的视频进行隐藏，删除，更改封面，标题等一系列操作，之后会进行重新上传更新。

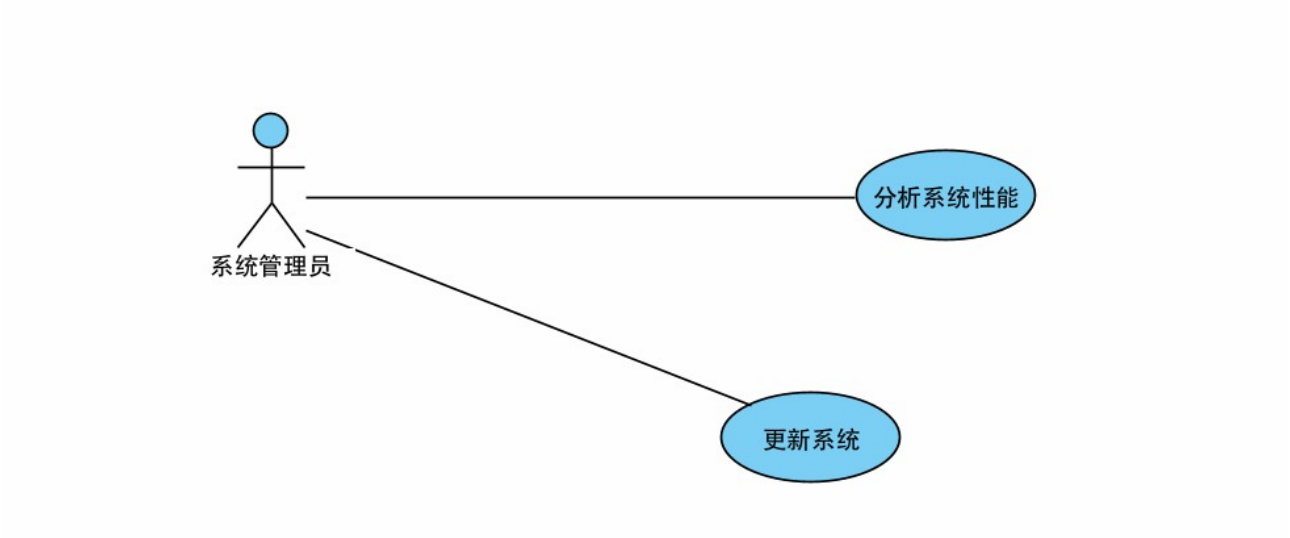
3.3.4 第四部分：**审核员**

审核员：审核员是负责审核正式用户所发出的各种各样的内容（包括弹幕评论，视频内容，视频，

动态）的人，拥有着驳回和同意视频或动态上传到首页或者动态朋友圈，删除评论或者弹幕的权力。

管理正式用户：管理正式用户的视频本身和视频的描述，决定通过或者驳回视频发布;管理视频的弹幕和评论，移出违规内容;管理正式用户的账户，处理违规行为，决定封禁或者解除封禁。

3.3.5 第五部分：**系统管理员**



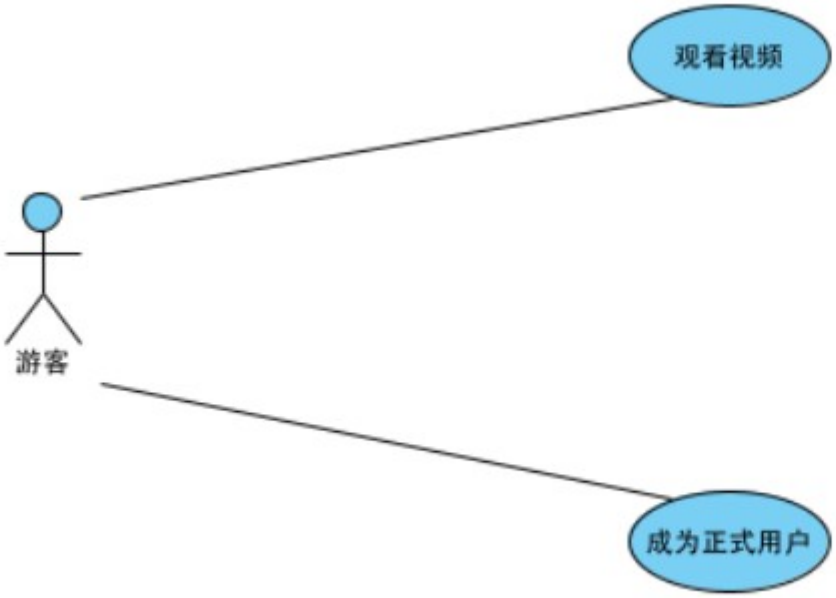
系统管理员：是负责整个系统的最高权限人，可以拥有配置系统全局参数的权限。

更新系统：系统管理员执行预定的更新流程，将更新的内容进行封装更新并发布。

分析系统性能：系统管理员分析系统性能，通过登录监控平台进行日常巡检，审视用户体验、应用服务、数据及基础设施等各层级的核心指标。

3.4. 用况的详细描述

3.4.1 第一部分：游客用况



3.4.1.1 观看视频

参与者：游客

后置条件：游客观看的对应视频观看次数发生变化
若通过搜索功能找到视频，搜索记录相关数据可能更新。

基本流：

1.游客打开软件或者访问应用首页，用况启动。

{搜索视频}

2.系统执行子流**<加载视频列表>**（S1）

{加载视频列表}

3.游客从展示的视频列表中选择一个公开的视频点击观看。

4.系统执行子流**<播放视频>**（S2）

{播放视频}

5.视频播放完毕或者游客关闭视频播放界面，用况终止。

备选流：

A1 **“搜索视频”**（普适备选流）

触发条件：游客在浏览首页视频列表后，未找到感兴趣的视频，或者有明确目标希望通过搜索快速定位特定视频。

1.系统响应游客的搜索操作，展示搜索页面，包含搜索框和可能的搜索历史记录（如果有）。

2.游客在搜索框中输入想要查找视频的关键词，然后点击搜索按钮。

3.系统执行子流**<根据关键词搜索视频>**（S3）

{执行搜索查询}

{显示搜索结果}

事件流恢复：事件流恢复到基本流第二步，系统尝试加载视频列表。

A2 **“处理视频列表加载失败”**（特定备选流）

扩展点及其触发条件：在**{加载视频列表}**扩展点上，当系统因为网络错误或者服务异常无法加载视频列表。

1.系统显示加载失败，请检查网络连接的提示。

2.系统提供重试的按钮

3.游客点击重试按钮

事件流恢复：事件流恢复到基本流第二步，系统再次尝试加载视频列表。

A3 **“处理视频播放错误”**（特定备选流）

扩展点及其触发条件：在**{播放视频}**扩展点上，当所选视频被删除或者格式不支持等原因导致无法播放

1.系统提示视频暂时无法播放。

2.系统自动返回视频列表页面。

3.浏览首页的公开视频用况终止。

A4 **“处理搜索服务异常”**（特定备选流）

扩展点及其触发条件：在 **{执行搜索查询}** 扩展点上面，当搜索服务不可用（网络不稳定或断网）或者查询出错的时候。

1.系统在页面上提示“搜索服务暂不可用，请稍后再试”。

2.搜索内容用况终止。

A5“处理空搜索结果”（特定备选流）

扩展点及其触发条件：在**{显示搜索结果}**扩展点上，当系统根据关键词查询之后，没有找到任何匹配的视频。

1.系统在搜索结果页面显示“视频找不到”的提示。

2.同时，系统在搜索结果页面显示一些热门视频推荐。

3.搜索内容的用况终止。

3.4.1.2 成为正式用户

参与者：游客

前置条件:系统终端网络连接正常。云数据库空间足够，且正常工作。

后置条件:游客成功转变为正式用户，系统自动更新用户信息，新用户数据存储到数据库中。

基本流：

1.游客点击**“成为正式用户”**按钮之后，用况开始启动。

2.系统显示与成为正式用户相关的界面，提供两个选项 - 直接注册成为正式用户和第三方登录成为正式用户选项。

{选择成为正式用户的方式}

3.游客选择直接注册成为正式用户方式，并完成信息提交。

{验证成为正式用户的信息}

4.系统执行子流**<验证信息>** (S4)。

5.系统自动登录，然后显示成为正式用户成功提示。

6.用况终止。

备选流

A1**“采用第三方账号登录成为正式用户”**（特定备选流）

扩展点及其触发条件：在**{选择成为正式用户的方式}**扩展点上，当游客选择第三方登录的时候。

1.系统根据游客选择的第三方平台，跳转至对应的第三方授权界面。

2.用户在第三方平台完成身份验证并且授权。

3.第三方系统将验证成功的凭证和用户基本信息返回给本系统。

4.系统接收基本信息，创建本地账户并且完成登录，游客正式成为本系统正式用户。

事件流恢复：接着基本流第 5 步显示成为正式用户成功提示之前，即系统完成登录后直接进入显示成为正式用户成功提示环节。

A2 **“处理无效成为正式用户的信息”**（特定备选流）

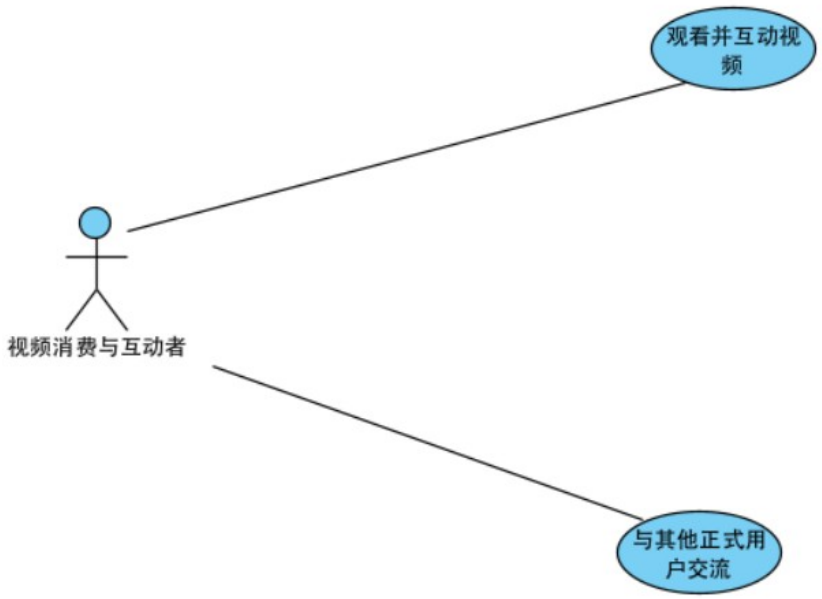
扩展点及其触发条件：在 **{验证成为正式用户的信息}** 扩展点上，当系统验证发现信息有误（比如用户名已经存在，格式错误等）。

1.系统在注册表单对应的位置提示具体的错误信息（如"该用户名已存在"）。

2.系统保留游客已填写的其他正确信息。

事件流恢复：恢复到基本流第 3 步的信息提交环节，游客根据提示修正错误信息后重新提交。

3.4.2 第二部分：视频消费与互动者用况



3.4.2.1 观看并互动视频

参与者：视频消费与互动者

后置条件：视频观看次数、评论等属性改变

基本流：

1.当参与者 **“视频消费互动者”** 进入系统并选择主页视频区域时，用况启动。

{搜索视频}

2. 系统执行子流**<加载视频列表>**（S1）

{加载视频列表}

3.**视频消费互动者**从系统展示的视频列表中选择一个公开的视频点击观看。

4.系统执行子流**<播放视频>**（S2），应用视频的的默认设置（如音量、分辨率）。

{播放视频}

5.**视频消费互动者**观看视频内容，系统记录观看进度。

{观看中}

6.视频播放完毕或者**视频消费互动者**选择结束当前视频窗口，用况终止。

备选流:

A1 **“搜索视频”**（普适备选流）

触发条件：任何时候视频消费互动者在浏览首页视频列表或正在观看视频时，未找到感兴趣的视频，或者有明确目标希望通过搜索快速定位特定视频。

1.系统响应视频消费互动者的搜索操作，展示搜索页面，包含搜索框和可能的搜索历史记录（如果有）。

{搜索结果}

2.**视频消费互动者**在搜索框中输入想要查找视频的关键词，然后点击搜索按钮。

3.系统执行子流**<根据关键词搜索视频>**（S3）

{执行搜索查询}

{显示搜索结果}

事件流恢复：事件流恢复到基本流第二步，系统尝试加载视频列表。

A2 **“处理视频列表加载失败”**（特定备选流）

扩展点及其触发条件：在**{加载视频列表}**扩展点上，当系统因为网络错误或者服务异常无法加载视频列表。

1.系统显示加载失败，请检查网络连接的提示。

2.系统提供重试的按钮

3.**视频消费互动者**点击重试按钮

事件流恢复：事件流恢复到基本流第 2 步，系统再次尝试加载视频列表。

A3 **“处理视频播放错误”**（特定备选流）

扩展点及其触发条件：在**{播放视频}**扩展点上，当所选视频被删除或者格式不支持等原因导致无法播放

1.系统提示视频暂时无法播放。

2.系统自动返回视频列表页面。

3.浏览首页的公开视频用况终止。

A4 **“处理搜索服务异常”**（特定备选流）

扩展点及其触发条件：在 **{执行搜索查询}** 扩展点上面，当搜索服务不可用（网络不稳定或断网）或者查询出错的时候。

1.系统在页面上提示“搜索服务暂不可用，请稍后再试”。

2.搜索内容用况终止。

A5 **“处理空搜索结果”**（特定备选流）

扩展点及其触发条件：在 **{显示搜索结果}** 扩展点上，当系统根据关键词查询之后，没有找到任何匹配的视频。

1.系统在搜索结果页面显示“视频找不到”的提示。

2.同时，系统在搜索结果页面显示一些热门视频推荐。

3.搜索内容的用况终止。

A6 **“处理互动 ”**（普适备选流）

触发条件：在**{观看中}**处的任何时间，当视频消费互动者对视频进行互动时。

{检查当前用户账户}

1.系统获取当前用户之前对该视频的互动数据。

2.系统根据用户互动类型来执行操作：

a:当用户选择点赞操作时：

a.1:如果之前用户已经点赞，取消点赞操作; 如果用户没有点赞，视频点赞数加一，显示点赞特效。

a.2:系统通知当前视频作者被点赞。

a.3:事件进入到基本流 3.

b:当用户选择收藏操作时：

b.1：如果当前用户对该视频已经收藏过，取消收藏操作，用户收藏夹取消该视频；如果没有收藏，当前视频的收藏加一，当前用户收藏夹内容添加该视频。

b.2：事件进入到基本流 3

c:当用户选择投币操作时：

{硬币数检查}

c.1：如果当前用户对该视频已经投币过，无法进行投币操作，如果没有投币，当前视频的投币数加一，用户硬币数量减少。

c.2：系统将用户投币数给当前视频作者，并提醒作者。

c.3：事件进入到基本流 3

d：当用户选择缓存操作时：

d.1：如果当前用户对该已经缓存过，无法进行缓存操作；如果没有缓存，系统获取视频数据，下载视频到用户设备中。

{下载中}

d.2：事件进入到基本流 3

e：当用户选择转发操作时：

e.1:系统获取当前视频的信息，生成视频链接。

e.2:系统提供窗口用于用户转发链接，用户能够复制当前链接。

e.3:事件进入到基本流 3

f：当用户选择评论操作时:

f.1: 系统获取当前视频的信息与评论信息，并且显示 。

f.2:用户能够输入文字，选择表情，点击发送按钮能够发送评论到当前视频的评论区中。

f.3:系统获取用户输入的消息，将数据同步到评论区中。

f.4:事件进入到基本流 3

g：当用户选择回复操作时：

g.1：系统获取当前视频的信息与评论信息，并且显示 。

g.2：用户能够输入文字，选择表情，点击发送按钮能够发送回复到当前视频评论的回复区域。

g.3:系统获取用户输入的消息，将数据同步到评论区中。

g.4:系统提醒被回复的用户。

g.5:事件进入到基本流 3

h:当用户选择对评论区的评论或回复点赞操作时：

h.1：系统获取当前视频的信息与评论信息，并且显示 。

h.2：用户对评论区中的评论或回复进行点赞，同时系统显示点赞特效。

h.3：系统获取用户点赞对象，将点赞消息通知给对应用户。

h.4：事件进入到基本流 3

i：当用户选择对评论区的评论或回复点踩操作时：

i.1：系统获取当前视频的信息与评论信息，并且显示 。

i.2 用户对评论区中的评论或回复进行点踩，同时系统显示点踩特效。

i.3：事件进入到基本流 3

3.事件恢复到**{观看中}**，执行子流**<更新系统视频属性>**（S5）。

A7"硬币数检查"（普适备选流）

触发条件：在**{硬币数检查}**拓展点上，当用户的硬币数量不够的时候。

1.系统对当前用户显示硬币数量不够的提示。

2.取消用户的投币操作。

3.事件恢复到**{观看中}**。

A8 **“下载处理”**（普适备选流）

触发条件：在**{下载中}**处的任何时间，当网络断开或者不佳的时候。

1.系统停止下载视频到用户。

2.系统提示用户因为网络问题下载视频失败。

3.事件恢复到**{观看中}**。

A9**“下载处理”**（普适备选流）

触发条件：在**{下载中}**处的任何时间，当本地设备存储空间不够的时候。

1.系统停止下载视频到用户。

2.系统提示用户存储空间不够，导致视频下载失败。

3.事件恢复到**{观看中}**。

A10**“账户检查”**（特定备选流）

触发条件：在**{检查当前用户账户}**拖着点上，当账户被封禁的时候。

1.系统查询账户详情，获取封禁原因和封禁期限。

2.系统向用户显示封禁提示信息，包括账户状态、封禁具体原因、封禁期限和解封方式。

3.系统阻止用户执行任何互动操作（如点赞、评论）。

4.事件恢复到**{观看中}**。

3.4.2.2 与其他正式用户交流

参与者：视频消费与互动者

前置条件：**视频消费与互动者**已成功登录系统;

私信与动态交流功能处于正常服务状态;
视频消费与互动者账户未被禁言或封禁;
交流目标用户（其他**正式用户**）账户状态正常且可接收信息;

后置条件：系统已记录交流内容;
系统已更新参与者交流状态;
系统已应用动态分享可见性;

基本流：
1.**视频消费与互动者**登录成功之后，用况启动。
{选择交流方式}
2.系统显示首页（包括私信和动态分享）。
3.系统保存交流内容（比如私信或者动态内容），更新视频消费与互动者的交流状态。
4.用况终止

备选流：
A1 **私信交流** (特定备选流)
扩展点及其触发条件：在**{选择交流方式}**处，当视频消费与互动者点击消息时：
1.系统弹出消息页面。
2.视频消费与互动者点击想要交流的对象目标。
3.系统弹出聊天历史记录提示输入私信内容。
{私信内容违规}
4.**视频消费与交互者**输入消息并发送给指定目标用户。
事件流恢复：视频消费与互动者点击关闭聊天界面后，事件流返回基本流第 3 步。

A2 **动态分享交流**（特定备选流）
扩展点及其触发条件：在**{选择交流方式}**处，当视频消费与交互者选择动态分享时：
1.系统提示参与者编写动态内容（如文本、媒体）。
2.**视频消费与交互者**输入内容并设置可见性（如公开或仅好友）。
{动态信息违规}
3.系统发布动态信息，并应用可见性规则。
事件流恢复：**视频消费与互动者**点击关闭动态界面后，事件流返回基本流第 3 步。

A3 **发送私信内容违规** （特定备选流）
扩展点及其触发条件：在备选流 A1**{私信内容违规}**处，当视频消费与互动者发送的私信内容违规时：
1.系统检测到违规内容，拦截发送的私信，阻止内容被保存和传递。
2.系统向用户显示明确的违规提示信息，提示简要说明违规类型。
3.**视频消费与交互者**修改内容后重新发送，系统将再次进行内容安全检测。
事件流恢复：**视频消费与互动者**发送成功后，返回到 A1 备选流第 4 步。

A4 **发送动态内容违规** （特定备选流）
扩展点及其触发条件：在备选流 **A2{动态信息违规}**处，当**视频消费与互动者**发布的动态内容违规时：
1.系统检测到动态内容违规，拦截发布请求。随后向用户显示明确的违规提示信息，并简要说明违规类

型。

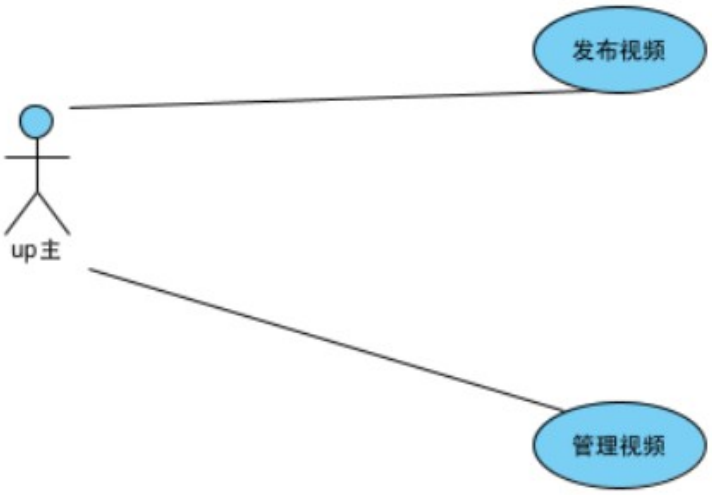
2.系统自动保留已输入的动态内容至编辑状态，方便用户直接修改。

3.**视频消费与交互者**对违规内容进行修改后，重新提交发布请求。

4.系统对修改后的内容再次进行安全检测。

事件流恢复：视频消费与交互者发布动态成功后，返回到 A2 备选流第 3 步。

3.4.3 第三部分：up 主



3.4.3.1 发布视频

参与者：up 主

前置条件：**up 主**账户正常

后置条件：视频记录创建，系统首页增加新视频，提醒粉丝新视频已更新

基本流：

1.**up 主**在首页点击上传视频按钮之后，用况开始启动

2.系统显示上传界面。

{选择与填写视频信息}

3.up 主选择视频文件，并且填写并且选择视频信息（比如标题，简介，关键词，封面等）

{执行上传}

4.**up 主**点击上传。系统接收视频文件和视频封面文件，并判断视频文件和视频封面文件的格式。

{视频文件格式正确}

5.系统将视频文件保存，并且保存视频信息和视频封面。

6.系统将视频信息保存中。

7.系统开始上传视频。

8.系统显示上传进度并进行文件上传。

9.视频上传成功，视频进入审核队列。

10.审核员审核视频文件及视频信息。

{审核视频}

11.审核员确认合规后，审核员同意发布该视频。

12.视频发布成功。

13.用况终止。

备选流：

A1“处理视频信息无效”（特定备选流）

扩展点及其触发条件：在**{选择与填写视频信息}**扩展点上，当视频信息不符合要求时（如标题为空等情况的时候）。

1.系统提示具体的错误信息（如请填写视频标题等）。

2.阻止进入上传步骤。

事件流恢复：事件流恢复到基本流第 3 步，**up 主**修改信息或重新选择文件。

A2“处理上传中断”（特定备选流）

扩展点及其触发条件：在**{执行上传}**扩展点上，当上传过程因网络等原因中断时。

1.系统提示上传中断。

2.提供重新上传或继续上传选项。

事件流恢复：用户选择操作后，事件流尝试恢复到基本流第 4 步。

A3“处理视频文件无效”（特定备选流）

扩展点及其触发条件：在**{视频文件格式正确}**扩展点上，当视频文件不符合要求时（如格式不支持、大小超限等情况的时候）。

1.系统提示具体的错误信息（如请格式不支持等）。

2.阻止进入上传步骤。

事件流恢复：事件流恢复到基本流第 3 步，**up 主**重新选择视频文件。

A4“审核不通过”（特定备选流）

扩展点及其触发条件：在**{审核视频}**扩展点上，当因视频文件或信息不合规时（如有违禁内容等情况的时候）。

1.审核员审核不通过，并输入原因。

2.系统将结果反馈给**up 主**。

3.系统删除上传的视频文件及视频信息。

事件流恢复 ：事件流恢复到基本流第 3 步，**up 主**重新选择视频文件或更改视频信息。

3.4.3.2 管理视频

参与者：up 主

前置条件：up 主账户正常、且视频存在

后置条件：视频信息更新

基本流：

1.**up 主**进入内容管理界面之后，用况开始启动。

2.系统获取当前用户。

3.系统获取当前用户上传的视频。

{加载视频列表}

4.系统显示该 up 主的视频列表。

5.**up 主**选择一个视频。

6.**up 主**执行管理视频相关操作（比如修改标题、封面，管理评论区，删除或者隐藏等操作）。

7.**up 主**提交管理内容。

8.系统将新的视频内容保存。

{处理稿件操作}

9.视频进入审核队列。

10.审核员审核稿件更改的相关内容。

{审核视频}

11.审核员确认合规后，审核员同意更改视频相关内容。

12.系统提示操作成功。

13.系统刷新，并且显示更新后的视频稿件列表。

14.用况终止。

备选流：

A1 “处理列表加载失败”（特定备选流）

扩展点及其触发条件：在**{加载稿件列表}**扩展点上，当稿件列表加载失败时。

1.系统提示加载失败。

事件流恢复：管理视频用况终止或提供重试机制然后重新回到基本流第 3 步。

A2“处理操作失败”（特定备选流）

扩展点及其触发条件：在**{处理稿件操作}**扩展点上，当对稿件的操作（如编辑、删除等）失败时。

1.系统提示操作失败（如删除失败，请重试）。

事件流恢复：事件流恢复到基本流第 4 步后，up 主可重新尝试操作。

A3“审核不通过”（特定备选流）

扩展点及其触发条件：在**{审核视频}**扩展点上，当因修改内容不合规时（如有违禁内容等情况的时候）。

1.审核员审核不通过，并输入原因。

2.系统将结果反馈给 up 主。

3.系统恢复原有数据，不进行修改。

事件流恢复：事件流恢复到基本流第 5 步，**up 主**可重新尝试操作。

3.4.4 第四部分：审核员

3.4.4.1 管理正式用户

参与者：审核员
后置条件：视频状态更新，账户状态更新

基本流：

1.审核员进入管理正式用户页面之后，用况启动。
2.管理正式用户页面显示“视频队列”，“弹幕，评论队列”，“正式用违规行为和申诉队列”。
3.审核员选择审核队列。
{选择审核队列}
4.审核员退出审核页面，用况终止。

备选流：

A1**“管理视频和视频信息”**（特定备选流）
扩展点及其触发条件：在**{选择审核队列}**处，选择“视频队列”。
1.系统将所有等待审核的视频以及视频信息加载到审核队列。
{加载视频详细}
2.审核员选中一个视频，系统播放审核员选中的视频并且显示它的详细信息。
3.审核员跟据系统提供的规则做出通过或者驳回决定，并且给出驳回理由和违规行为。
{做出审核决定}
4.系统根据审核结果，将视频状态更新为 up 主想要的视频状态或者驳回给 up 主，并记录违规行。
5.系统将这个视频和视频信息移出“视频队列”。
事件流恢复：恢复到基本流第 3 步。

A2**“管理视频的弹幕和评论”**（特定备选流）
扩展点及其触发条件：在**{选择审核队列}**处，选择“弹幕，评论队列”。
1.系统将所有弹幕和评论加载到“弹幕，评论队列”中。
2.审核员根据用户举报和系统提供的审核弹幕/评论规则，决定删除或者保留。
{执行审核操作}
3.系统根据审核员对弹幕和评论的审核结果，如果违规，从平台移出违规内容，并记录违规行为。
4.系统将这个弹幕和评论移出“弹幕，评论队列”。
事件流恢复：恢复到基本流第 3 步。

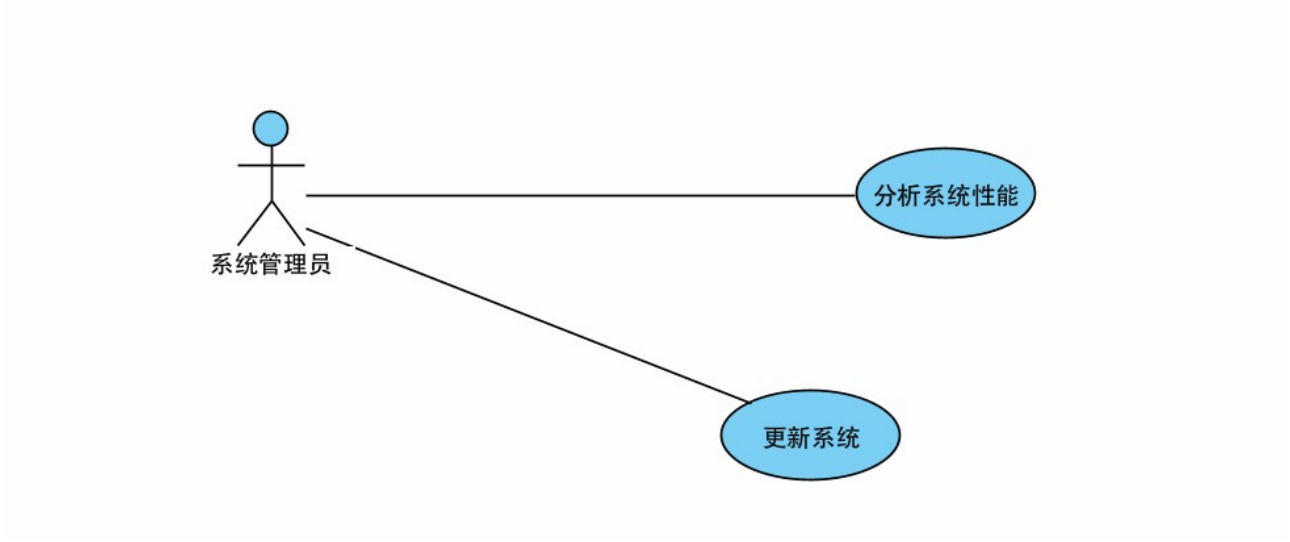
A3“管理正式用户账户”（特定备选流）
扩展点及其触发条件：在**{选择审核队列}**处，选择“正式用违规行为和申诉队列”。
1.系统将所有用户的违规行为和申诉加载到“正式用违规行为和申诉队列”中。
2.审核员从“正式用违规行为和申诉队列”中选择一个用户，系统显示该用户的详细信息。
3.审核员根据平台规则决定将选中的账户封禁或者解封或者给予警告。
4.系统将这个账户移出“正式用违规行为和申诉队列”。
4.系统更新账户状态。
事件流恢复：恢复到基本流第 3 步。

A4“标记为稍后审核”（有界备选流）
扩展点及其触发条件：在**{加载视频详情}**和**{做出审核决定}**扩展点之间的任何位置，当审核员暂无法做出决定时。
1.审核员选择稍后再审。
2.系统将该视频标记，并返回待审核队列。
事件流恢复：恢复到备选流 A1 第 2 步。

A5“处理审核操作失败”（特定备选流）
扩展点及其触发条件：在**{做出审核决定}**扩展点之后，当系统执行审核操作（更新状态）失败时（如视频已被其他审核员处理）。
1.系统提示操作失败或视频已被处理。
2.如果视频已被其他审核员处理，系统自动刷新审核队列，移出该视频。
事件流恢复:备选流 A1 第 2 步。

A6“处理审核操作失败”（特定备选流）
扩展点及其触发条件：在**{执行审核操作}**扩展点上，当操作执行失败的时候。
1.系统提示操作失败
事件流恢复:重试恢复到备选流 A2 第 2 步。

3.4.5 第五部分：系统管理员



3.4.5.1 分析系统性能

参与者：系统管理员

前置条件：系统管理员通过身份验证并登陆到监控系统;
该系统管理员具有访问关键性能数据的权限;
监控系统监控系统此时运行正常，提供的数据实时并准确;

后置条件:
成功路径:
此次巡检已被系统记录;
问题进入了处理状态;
失败路径:
系统性能状态因分析过程中断而无法确定;
监控数据源连接超时被记录;

基本流:
{系统显示警告}A1
1：管理员登陆统一监控平台,用况启动
{选择性能数据}A3
2：系统展示性能数据
{检查性能指标}A4
3：管理员按照架构层次检查核心性能指标，确认一切正常
4：用况终止

备选流:

A1：响应外部性能警告（有界备选流）
扩展点及触发点： 在**{系统显示警告}和{选择性能数据}**之间，当监控系统产生一条高优先级的性能告警并通知给管理员时，立即中断任何当前活动（包括基本流）并触发
1：系统管理员收到监控系统的警告通知
2：管理员查看警告
3：定位问题页面
{查看日志内容}A2
4：管理员分析日志数据
5：确定问题根源
事件流恢复：当系统管理员找到问题根源并关闭问题页面后，事件流返回到基本流 2

A2：选择系统日志
扩展点及触发点：在**{查看日志内容}**处，当系统管理员想要查看日志时，需要在选择想查看指定日期的日志时，显示内容。
1.管理员选择日期
2.系统判断当前管理员是否有对应权限查看日志
3.显示日志上下文
事件流恢复：系统管理员关闭日志，事件流从暂停点继续

A3:选择不同层级数据（特定备选流）
扩展点及触发点：在{选择性能数据}时，当系统管理员需要选择不同层级数据（用户体验或应用服务等）
1.系统管理员选择需要查看的层级数据
2.系统检查当前系统管理员是否有足够权限可以查看此层级数据
3.系统加载数据面板
事件流恢复：系统加载处管理员所需要数据，回到基本流 3

A4：发现并调查资源瓶颈（特定备选流）
扩展点及触发点： 在{检查性能指标}处，当管理员观察到一个或多个性能指标值超出预设的正常阈值 时，此备选流被触发
1：系统管理员分析该异常指标及其关联指标
2：系统管理员定位问题的根本原因
事件流恢复：回到基本流 3 继续检查指标，或者触发新的用况（解决问题）

3.4.5.2 更新系统

参与者：系统操作员

前置条件：当前系统有功能或性能上有提升或者更改的需要，或系统存在部分 Bug 需要进行修复
后置条件：系统组件版本更新，或系统成功回滚到上一个稳定版本。

基本流：
1. 系统操作员登录运维管理平台并进入“系统更新”模块后，用况启动。
2. 系统显示更新仪表盘，包括“准备更新包”、“执行服务更新”、“执行数据库更新”、“监控与验证”等主要步骤。
3. 系统操作员按顺序执行更新流程。
   {执行更新流程}
4. 系统操作员确认所有更新步骤完成且系统稳定，用况终止。

备选流：

A1 **“执行后端服务更新”**（特定备选流）
扩展点及其触发条件： 在**{执行更新流程}**处，选择“执行服务更新”。
1. 系统操作员选择预先上传的应用程序包（如 Docker 镜像）和目标服务器集群。
2. 系统启动发布流程，在负载均衡器后逐步将流量切换到运行新版本服务的节点。
3. 系统操作员监控新版本节点的运行状态（如 CPU、内存、错误日志）。
   {监控新服务状态}
4. 当所有流量都成功切换到新版本节点且运行稳定后，系统回收旧版本节点的资源。
5. 系统将“执行服务更新”步骤标记为完成，用况终止。

A2 **“执行数据更新”**（特定备选流）
扩展点及其触发条件：在**{执行更新流程}**处，选择“执行数据库更新”。
1. 系统操作员上传或选择需要执行的数据库更新脚本。
2. 系统首先在备份数据库上执行模拟运行，并生成影响报告。
   {执行更新脚本}
3. 系统操作员确认报告无误后，批准在生产数据库上执行脚本。
4. 系统执行更新脚本，并记录所有变更。
5. 系统将“执行数据库更新”步骤标记为完成。

A3 **“执行回滚操作”**（特定备选流）
扩展点及其触发条件：在{监控新服务状态}扩展点之后，当系统监控到关键指标（如错误率、响应时间）严重超标时。
1. 系统操作员在运维平台触发“一键回滚”操作。
2. 系统自动将负载均衡器的流量全部切回至旧版本的服务节点。
3. 如果更新包含数据库变更，系统执行预先生成的数据库回滚脚本，将数据恢复到更新前的状态。
4. 用况终止。

A4 **“暂停与继续更新”**（有界备选流）
扩展点及其触发条件：在{执行更新流程}中的任何步骤，当需要临时暂停以进行人工检查或等待指令时。
1. 系统操作员选择“暂停更新”。
2. 系统暂停当前自动化流程，并保持当前状态。系统在仪表盘上高亮显示“更新已暂停”。
3. 当系统操作员选择“继续更新”后，**事件流从暂停点恢复执行**。

A5 **“处理更新前置检查失败”**（特定备选流）
扩展点及其触发条件：在用况启动后，执行基本流第 3 步之前，当系统的健康检查或资源检查未通过时。
1. 系统在更新仪表盘上显示检查失败警告，并列出具体失败项（如：磁盘空间不足、存在未恢复的故障节点）。
2. 更新流程被阻止启动。
3. 系统操作员必须解决前置检查失败的问题后，才能重新启动更新流程。**事件流恢复**到基本流第 1 步。

A6 **“处理更新包验证失败”**（特定备选流）

扩展点及其触发条件：在**备选流 A1 第 1 步之后**，当系统检测到更新包的数字签名无效或完整性校验失败时。
1. 系统提示“更新包验证失败”，并中止部署流程。
2. 系统操作员需要重新获取有效的更新包并重新上传。
事件流恢复：恢复到备选流 A1 第 1 步。

3.5. 子流

3.5.1 S1 加载视频列表

1.系统向服务器发送请求，获取各个 **up 主**公开的视频列表数据。
2.服务器返回视频列表数据，系统接收并解析。
3.系统将解析后的视频列表展示在首页界面

3.5.2 S2 播放视频

1.系统获取游客点击的视频标识信息。
2.系统根据该标识信息向服务器请求视频播放数据。
3.服务器返回视频播放数据，系统接收并在视频播放界面进行播放。
4.用户点击“倍数，清晰度”。系统立即设置相应的播放。
5.系统在视频播放界面展示互动功能区（如点赞、评论、分享按钮等，游客点击后会弹出注册界面）。

3.5.3 S3 根据关键词搜索视频

前置条件：
游客已进入搜索页面。
网络连接正常，能够与服务器进行数据交互。
后置条件：
搜索结果成功展示，包含符合关键词的视频封面、标题、简介等信息。
系统记录本次搜索操作的时间、关键词、游客 ID 等信息（可选，根据需求记录）。
事件流：
1.系统将游客输入的关键词记录并发送。
2.服务器接收关键词并进行匹配。
3.服务器将搜索结果返回给系统。
4.系统接收搜索结果并解析。

3.5.4 S4 验证消息

1.系统获取游客输入的所有注册消息
2.系统检查输入的消息格式是否正确
3.系统检查当前输入的用户名是否已经被占用
4.如果信息有效，系统将注册信息保存

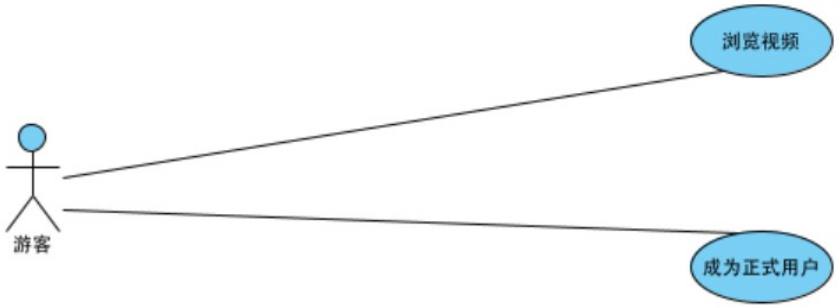
3.5.5 S5 更新系统视频属性

1.系统获取当前视频的全部属性，如点赞、评论数量。
2.系统将当前视频的全部属性覆盖原来视频的属性。
3.系统更新服务器中的视频数据。

4 需求分析

4.1. 健壮性分析

4.1.1 游客



对“观看视频”用况进行健壮性分析：

观看视频：游客无需登录，通过首页浏览或关键词搜索，加载并播放公开视频，包含视频列表加载、搜索、播放等子流。

首先确定边界类：游客用况涉及的边界有视频展示界面（MainPageUI）一个边界类。

确定控制类：处理视频浏览与播放的控制器为（VideoController）。

确定实体类：存储游客临时信息的实体为（Visitor），存储视频数据的实体为（Video）。

对“成为正式用户”用况进行健壮性分析：

成为正式用户包含：游客通过注册表单填写信息，提交信息后完成验证，转化为具备互动权限的正式用户。

首先确定边界类：注册登录界面（RegisterLoginUI）一个边界类。

确定控制类：处理注册登录流程的控制器为（ManagerUserController）。

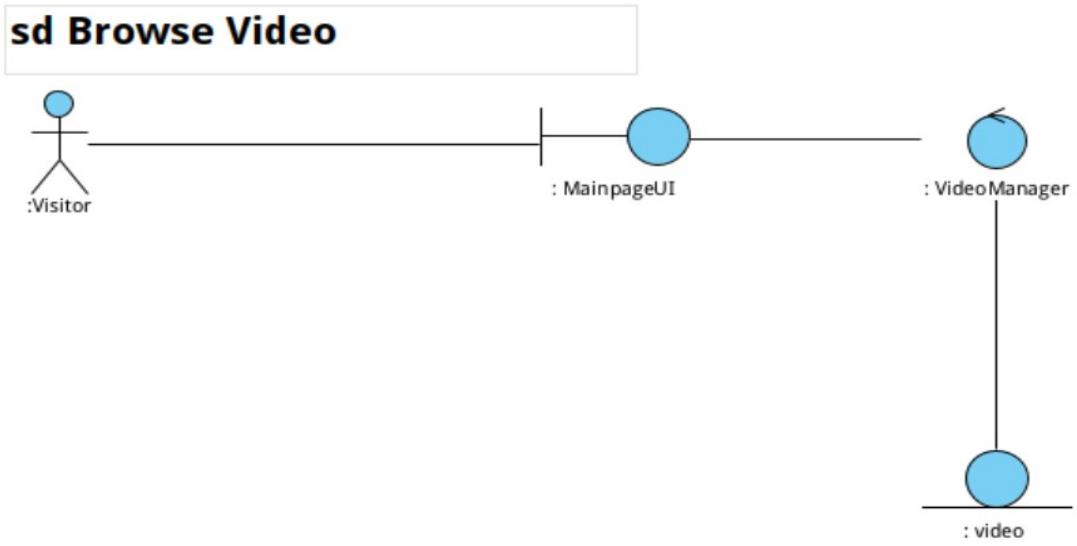
确定实体类：存储游客临时信息的实体为（Visitor），存储正式用户信息的实体为（VerifiedUser）。

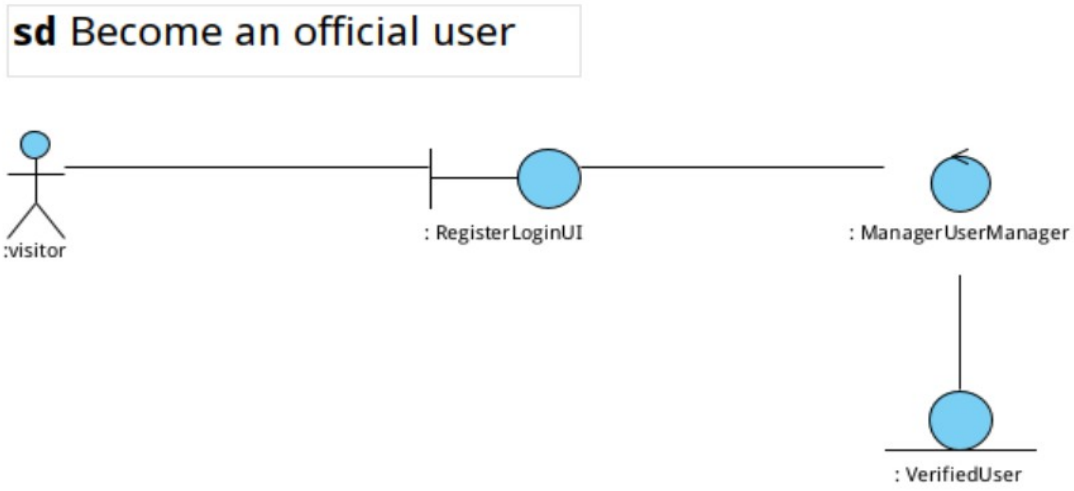
4.2. 交互建模

4.2.1 游客

4.2.1.1 协作图

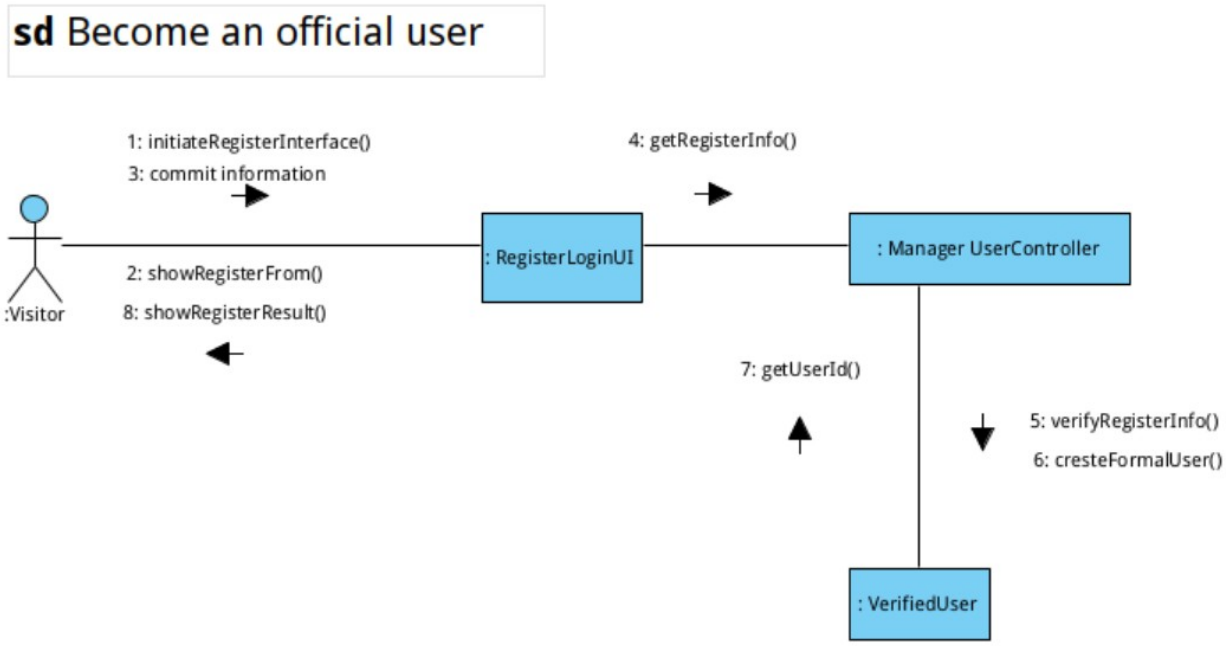
根据 4.1.1 中对游客相关用况进行健壮性分析，已经初步确定了主要参与的实体对象，与其连接的边界类和控制类，见下图。

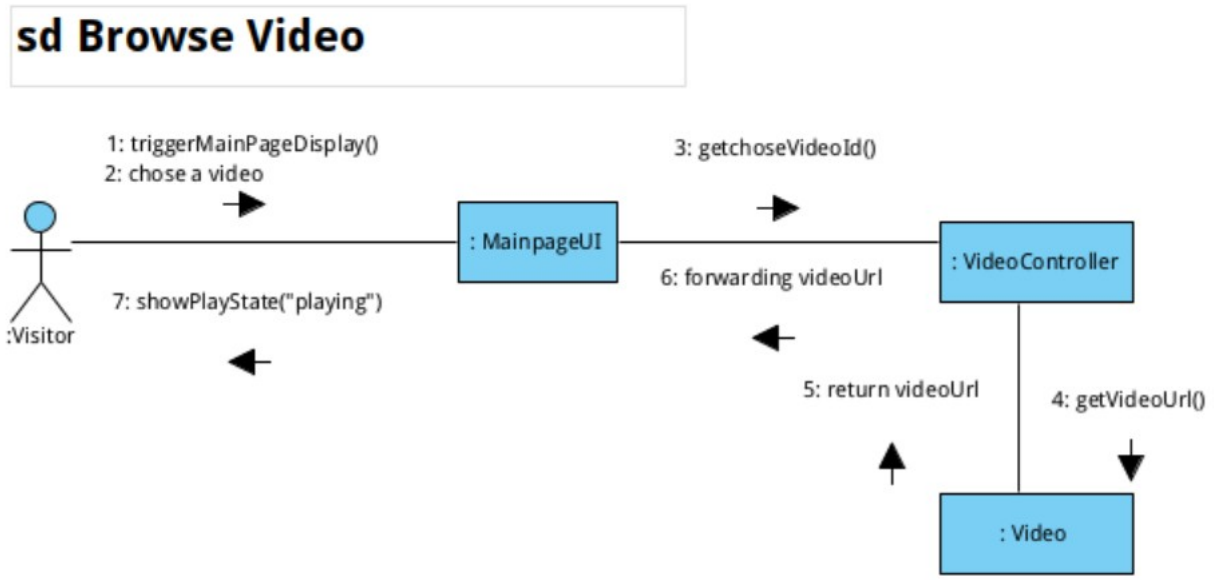




4.2.1.2 通信图

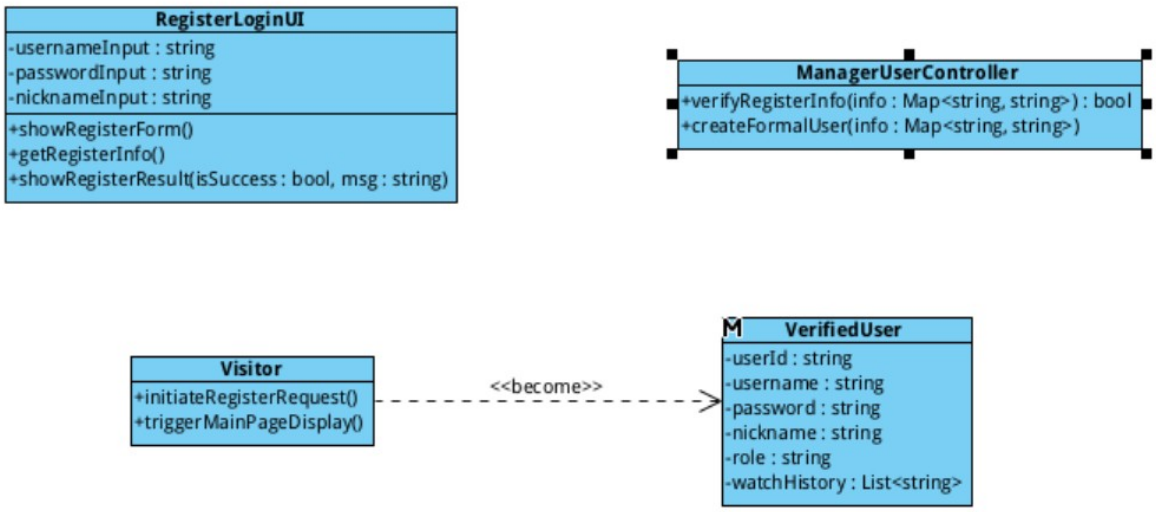
现在来绘制协助的通讯图：加入通讯图的边框，及在各个实体类、边界类、控制类之间的消息与连接，见下图。



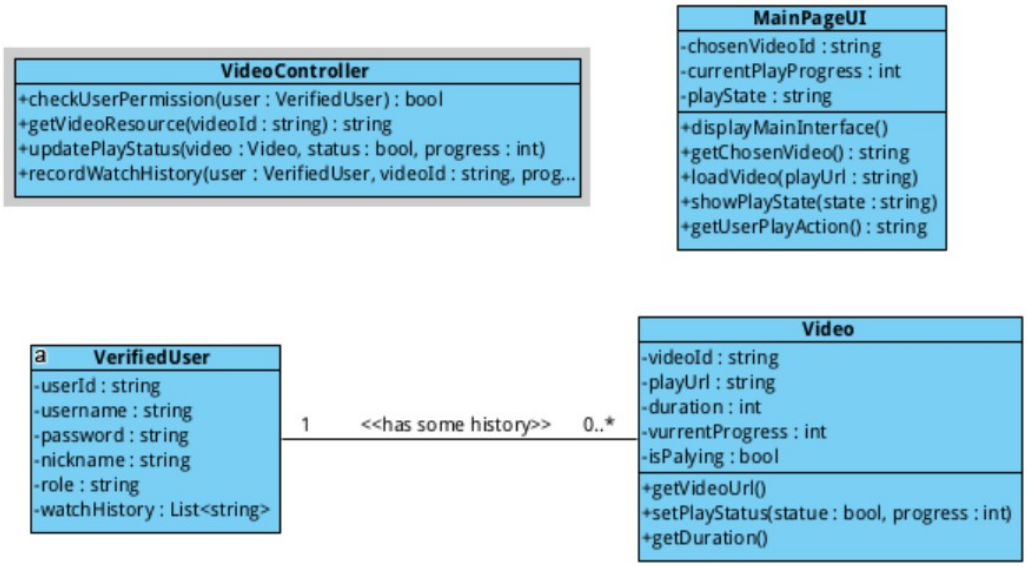


4.2.1.3 类图

根据 4.2.1.2 中的通讯图，生成对应于用况“成为正式用户”的类图，见下图。

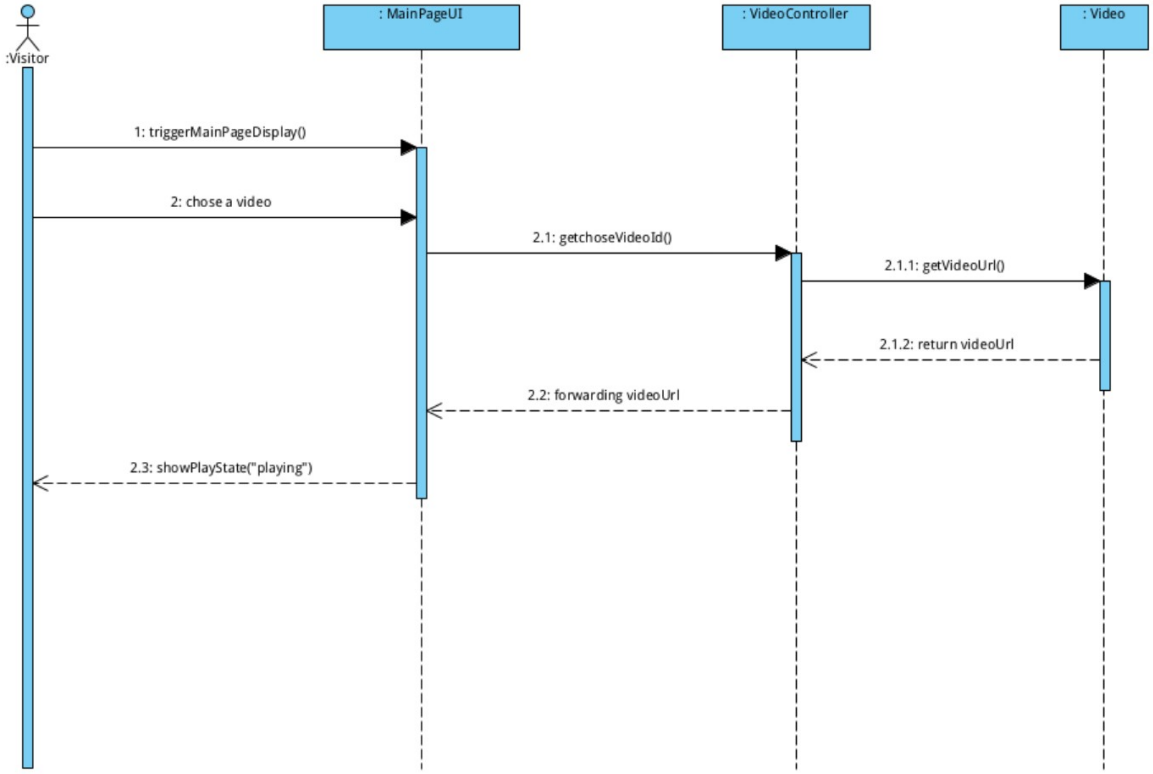


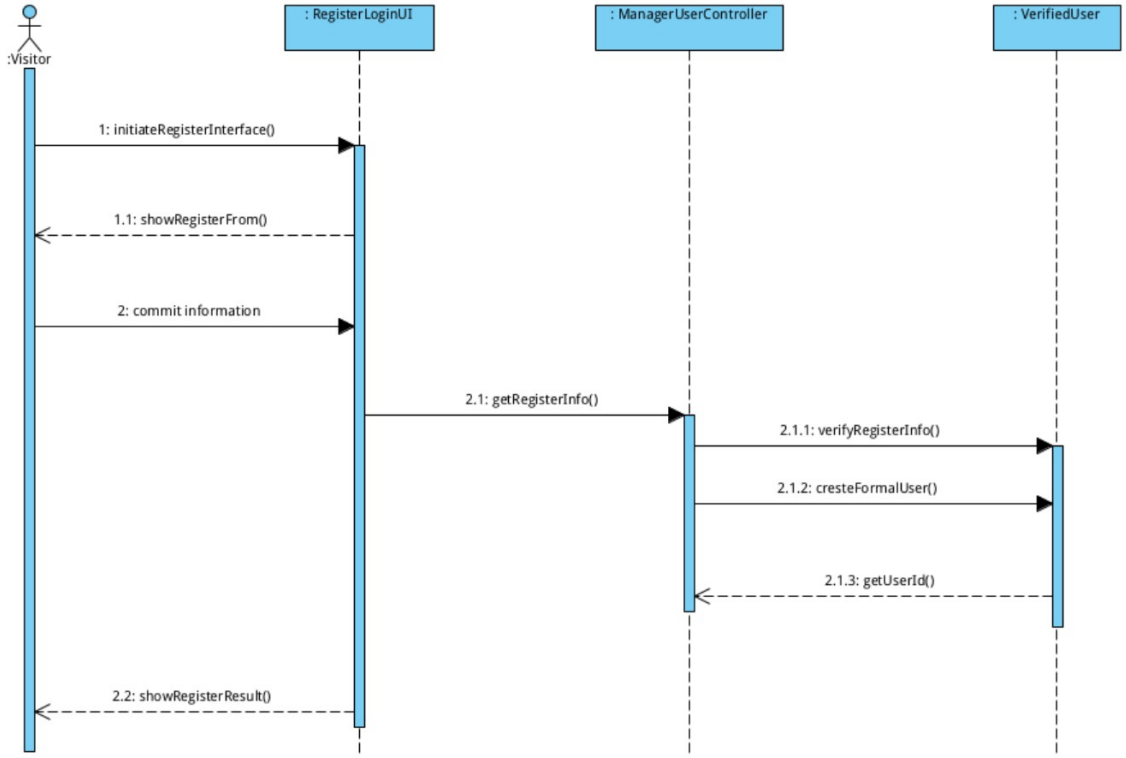
根据 4.2.1.2 中的通讯图，生成对应于用况“观看视频”的类图，见下图。



4.2.1.4 顺序图

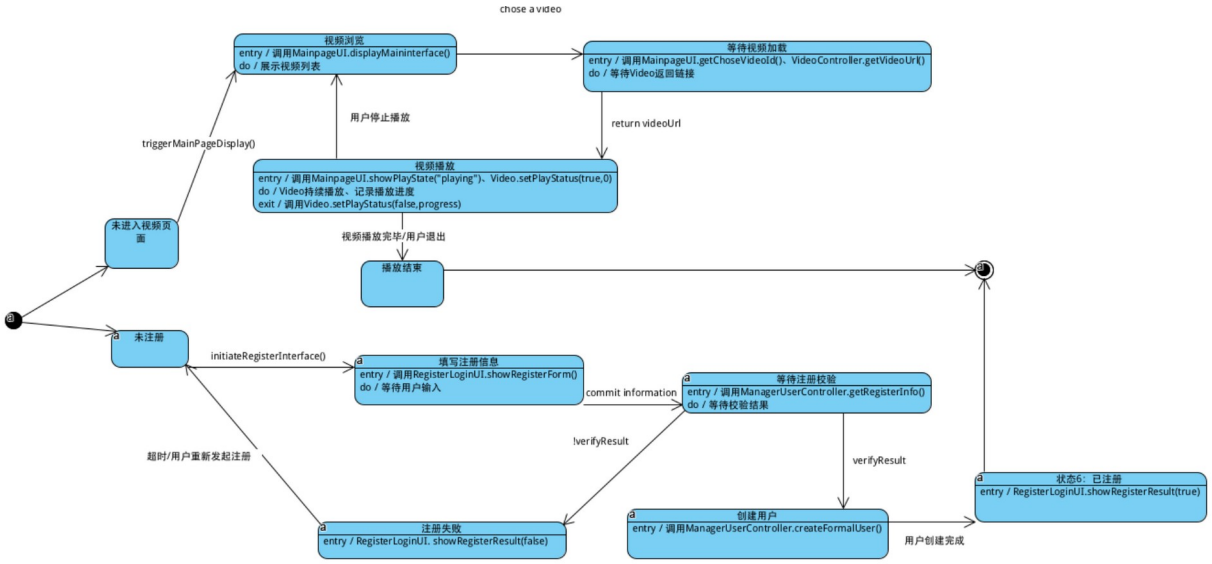
根据前述内容，我们绘制了关于用况“观看视频”、用况“成为正式用户”的带有边界和控制对象的顺序图，见下图。





4.2.1.5 状态机

下面我们用状态机来表示单个 Visitor 对象对于它所有参与的用况的所有响应，见下图。


chose a video
视频浏览
entry / 调用MainpageUI.displayMaininterface()
do / 展示视频列表
等待视频加载
entry / 调用MainpageUI.getChoseVideoId()、VideoController.getVideoUrl()
do / 等待Video返回链接
用户停止播放
return videoUrl
triggerMainPageDisplay()
视频播放
entry / 调用MainpageUI.showPlayState("playing")、Video.setPlayStatus(true,0)
do / Video持续播放、记录播放进度
exit / 调用Video.setPlayStatus(false,progress)
未进入视频页面
视频播放完毕/用户退出
播放结束
未注册
initiateRegisterInterface()
填写注册信息
entry / 调用RegisterLoginUI.showRegisterForm()
do / 等待用户输入
commit information
等待注册校验
entry / 调用ManagerUserController.getRegisterInfo()
do / 等待校验结果
!verifyResult
verifyResult
超时/用户重新发起注册
注册失败
entry / RegisterLoginUI. showRegisterResult(false)
创建用户
entry / 调用ManagerUserController.createFormalUser()
用户创建完成
状态6：已注册
entry / RegisterLoginUI.showRegisterResult(true)

5 架构设计

5.1. 设计目标与约束

5.1.1 设计目标

功能性目标：全面覆盖平台核心场景，包括游客浏览视频、正式用户互动交流、UP 主创作管理、审核员内容审核及系统管理员运维监控等全流程，确保各角色功能模块既独立运行又协同高效，精准落地“高质量视频 + 实时弹幕 + 深度社群” 的核心产品定位。

性能目标：适配高并发访问场景，严格达成首页首屏加载时间<3 秒、视频缓冲时间<5 秒的既定指标；保障弹幕实时推送、视频流畅播放及多用户同步互动，核心服务响应延迟控制在 100ms 以内，提升用户使用体验。

可靠性目标：确保系统 7×24 小时稳定运行，服务可用性不低于 99.9%；

安全性目标：实现用户密码加密存储、敏感操作二次验证；结合自动检测与人工审核机制拦截违规内容，防范合规风险；严格划分用户权限边界，避免数据泄露与非法操作。

可维护性目标：架构设计简洁清晰，明确各服务职责边界，支持独立部署与升级；配套完善的监控与日志系统，便于快速定位和排查问题；选用成熟稳定的技术栈，降低长期维护成本。

5.1.2 设计约束

技术栈约束：基于项目既定技术选型，后端采用 C++/QML，数据库使用 MySQL，前端选用 QML，需在该技术栈范围内实现架构落地，确保开发效率与系统兼容性。

成本约束：首阶段聚焦 PC 端核心功能，成本需可控，架构设计需避免过度设计，优先满足最小可行产品需求；云服务器、存储等资源配置需兼顾性能与经济性，支持按需扩容。

合规约束：需符合法律法规及监管要求，架构中需嵌入内容审核流程、青少年保护机制，确保用户数据收集与使用合规，版权内容管理规范。

多端适配约束：预留移动端扩展接口，架构设计需考虑 PC 端与后续移动端的数据同步与功能兼容，确保多端用户体验一致，数据互通无壁垒。

用户规模约束：初期聚焦 16-28 岁 Z 世代垂直用户群体，架构需支撑初期用户规模平滑增长，同时具备向大规模用户拓展的横向扩展能力，避免架构重构。

5.2. 系统总体架构

5.2.1 架构概览图

5.2.2 核心服务划分

服务名称	服务职责	关键技术	是否可以进行并发
视频浏览与播放服务	支撑游客及正式用户的视频列表加载、关键词搜索、视频播放（含倍速、清晰度调节）功能；记录视频观看次数与观看进度；保障视频流稳定传输。	MySQL 数据库查询优化、CDN 加速、子流 S1（加载视频列表）、S2（播放视频）、S3（关键词搜索视频）实现	是，支持多用户同时加载视频列表、搜索视频及播放不同视频
用户认证与管理服务	处理游客注册成为正式用	密码加密算法、第三方	是，支持多用户同时注

	户（含直接注册、第三方登录）、用户登录验证、个人信息管理；管控用户权限（如正常用户、UP 主、审核员、系统管理员）；支持用户账户封禁与解封操作。	登录授权接口、子流 S4（验证消息）	册、登录及权限校验
互动交流服务	支撑正式用户的视频互动（点赞、投币、收藏、评论、回复）、弹幕发送与展示、私信交流、动态发布与分享功能；实时同步互动数据与消息通知。	WebSocket（弹幕实时推送）、消息队列、子流 S5（更新系统视频属性）、违规内容初步检测	是，弹幕与互动操作采用异步处理
UP 主创作管理服务	支持 UP 主视频上传、视频信息（标题、封面、关键词）编辑、稿件管理（隐藏、删除、更新）；提供视频数据统计反馈（播放量、粉丝数、互动数据）；处理视频上传后的审核队列提交。	分片上传技术、文件存储服务（如对象存储 COS）、MySQL 事务管理、视频格式校验	是，支持多 UP 主同时上传视频与管理稿件，上传任务异步处理，文件存储与业务逻辑解耦提升并发量
内容审核服务	接收 UP 主上传视频、用户发布的弹幕 / 评论 / 动态等内容，提供审核队列管理；支持自动关键词检测违规内容、人工审核操作（通过 / 驳回 / 删除）；记录违规行为与审核结果。	文本关键词匹配算法、视频内容预览技术、审核流程引擎	是，审核队列按优先级调度，自动审核与人工审核并行处理，支持多审核员同时在线操作
系统运维监控服务	支撑系统管理员的系统性能分析（服务器状态、流量峰值、核心指标监控）、系统更新（服务更新、数据库更新）、全局参数配置、用户权限管控；提供日志查询与故障预警功能。	数据库备份与回滚技术	是，性能监控与系统更新独立执行，支持管理员同时进行多维度监控与运维操作

5.3. 分层架构设计

5.3.1 表现层

表现层作为系统与用户直接交互的窗口，核心职责为接收用户指令、渲染界面内容，并将用户请求委派给应用逻辑层。本层采用 QML 技术构建跨平台客户端，QML 的声明式语法与数据绑定特性，能够高效实现符合 Z 世代审美的动态、流畅交互界面，为弹幕互动、实时更新等核心场景提供技术支撑。

职责：专注于用户界面的渲染与用户交互事件的捕获，确保操作的即时反馈与视觉呈现的流畅性。

协作：表现层不包含业务逻辑，通过调用应用逻辑层提供的服务接口来完成业务请求，并依据返回数据更新界面状态。

5.3.1.1 前端界面

表现层包含以下核心用户界面，共同构成完整的用户交互体验：

登陆界面:负责用户身份认证入口，支持用户名/密码与第三方账号登录，是用户访问系统个性化功能的第一道门户。

首页界面（推荐，热门，影视，搜索）:作为内容分发的核心枢纽，集成推荐视频流、热门榜单、分区导航及全局搜索功能，旨在最大化用户的内容发现效率。

上传视频界面（上传文件，上传视频封面，标题，简介）:为 UP 主提供便捷的内容发布入口，支持视频文件与封面选择、标题与描述编辑，并展示上传进度。

视频播放界面（进度条，设置，弹幕设置，音量调节，分辨率设置，视频信息，评论区）:核心视频消费场景，提供视频播放、清晰度切换、弹幕渲染、进度控制以及点赞、评论、投币等即时互动操作面板。

消息界面（用户消息，私聊消息，朋友圈消息，系统消息）:整合系统通知、私信会话与动态流，构建用户间的沟通与信息获取渠道。

个人中心界面（个人名字，签名，历史记录，粉丝数，硬币数）:集中展示用户个人信息、创作数据（粉丝数、硬币数）、历史记录及收藏内容，是用户管理自身数字身份的主阵地。

审核界面:为审核员提供高效的内容管理工作站，以队列形式呈现待审核的视频、评论及用户报告，并附有一键通过/驳回操作。

5.3.2 应用逻辑层

职责：本身不实现业务规则，是业务流程的组织者（调用领域层）

5.3.2.1 主要服务

1. VideoService 视频服务

服务职责：作为视频内容管理的核心服务，负责视频上传、审核状态管理、元数据获取、播放控制及生命周期管理。

核心方法：

uploadVideo(videoFile, metadata)：接收视频文件与元数据（标题、描述等），验证文件格式与大小，暂存文件并触发审核流程（调用审核子系统）。

approveVideo(videoId, auditorId)：处理审核通过逻辑，更新视频状态为“已发布”，通知 UP 主（通过 MessageService），并索引视频（调用 SearchService）。

rejectVideo(videoId, reason)：处理审核驳回，记录驳回原因，通知 UP 主并允许重新编辑。

getVideoInfo(videoId)：返回视频元数据（如时长、标题、作者、播放量、互动数据），支持缓存优化。

playVideo(videoId, userId)：协调视频流传输（调用基础设施层），记录播放历史（用户行为用于推荐），更新播放计数。

updateVideoMetadata(videoId, updates)：允许 UP 主编辑视频标题、封面等，需验证权限。

deleteVideo(videoId)：软删除视频，更新状态并清理关联数据（如弹幕、评论）。

协作关系：

依赖审核子系统处理审核流程（通过事件驱动）。

与 UserService 验证 UP 主权限。

与 SearchService 同步视频索引。

通过 ResourceManagerController 管理视频文件存储。

2. UserService 用户服务

服务职责：管理用户身份认证、资料维护及社交关系（关注系统）。

核心方法：

register(username, password, profile)：验证用户名唯一性，加密存储密码，初始化用户资料（默认身份为 VerifiedUser）。

login(credentials, authType)：支持账号密码登录。

editProfile(userId, updates)：更新用户昵称、头像等，审计修改记录。

followUser(followerId, followeeId)：建立关注关系，更新粉丝数，触发消息通知（如通过 MessageService）。

unfollowUser(followerId, followeeId)：解除关注关系，同步更新计数。

getUserProfile(userId)：返回用户公开资料及统计信息（如视频数、粉丝数）。

协作关系：

与 RecommendationService 共享用户行为数据（如关注关系）。

依赖数据访问层持久化用户数据。

3. SearchService 搜索服务

服务职责：提供视频和用户的全文搜索与结果排序。

核心方法：

searchVideos(query, filters, sortBy)：基于关键词搜索视频，支持按分类、时长等过滤，按相关性、热度排序。

searchUsers(query, filters)：根据昵称或简介搜索用户，按粉丝数排序。

indexVideo(videoId, metadata)：当视频发布时，将视频元数据加入搜索索引（如 Elasticsearch）。

indexUser(userId, profile)：用户资料变更时更新索引。

协作关系：

依赖 RecommendationService 提供热门结果作为搜索降级策略。

与数据访问层交互获取详细数据。

4. RecommendService 推荐服务

服务职责：生成个性化视频推荐流，基于用户行为、内容特征及社交图谱。

核心方法：

generatePersonalizedFeed(userId, context)：综合用户历史行为（观看、点赞）、视频标签、好友动态生成推荐列表。

updateUserPreference(userId, event)：实时处理用户互动（如点赞、收藏），更新用户兴趣模型。

getTrendingVideos(limit)：返回全局热门视频，用于新用户冷启动。

refreshModel()：定时重训练推荐模型（离线任务）。

协作关系：

从 InteractionService 消费用户行为事件。

依赖 SearchService 进行候选视频检索。

5. MessageService 信息服务

服务职责：处理点对点私信、朋友圈动态及系统通知的收发与存储。

核心方法：

sendPrivateMessage(senderId, receiverId, content)：验证双方关系（如是否互关），存储消息并实时推送（WebSocket）。

getMessageThread(user1Id, user2Id)：返回私信会话历史。

postFriendCircleMessage(userId, content, visibility)：发布动态，应用可见性规则（如仅好友可见），触发好友通知。

likeMessage(messageId, userId)：处理动态点赞，更新计数。

sendSystemNotification(userId, template, data)：发送系统消息（如审核结果），支持模板化内容。

协作关系：

与 UserService 校验用户关系。

依赖基础设施层实现实时推送。

6. InteractionService 互动服务

服务职责：统一处理用户对视频的轻量级互动（点赞、收藏、分享、投币）。

核心方法：

likeVideo(userId, videoId)：记录点赞，更新视频点赞数，防止重复操作。

collectVideo(userId, videoId)：添加视频到用户收藏夹。

shareVideo(userId, videoId, platform)：生成分享链接，记录分享统计。

coinVideo(userId, videoId, amount)：验证用户硬币余额，扣减并更新视频投币数，通知 UP 主。

downloadVideo(userId, videoId)：验证下载权限（如大会员限制），生成临时下载链接。

协作关系：

与 UserService 校验用户状态（如封禁判断）。

与 VideoService 同步更新视频互动计数。

7. BarrageService 弹幕服务

服务职责：管理视频弹幕的实时发送、渲染与过滤。

核心方法：

sendBarrage(userId, videoId, content, timestamp)：验证弹幕内容（长度、敏感词），存储弹幕并广播给当前观看者。

getBarrages(videoId, startTime, endTime)：根据时间戳范围获取弹幕列表。

deleteBarrage(barrageId, operatorId)：用户删除自有弹幕或管理员删除违规弹幕。

likeBarrage(userId, barrageId)：记录弹幕点赞，更新热度。

协作关系：

依赖审核子系统进行弹幕内容过滤（前置检查）。

通过实时消息服务推送弹幕。

8. CommentService 评论服务

服务职责：管理视频评论的发布、审核、互动及树形结构维护。

核心方法：

addComment(userId, videoId, content, parentId)：发布评论或回复，初筛敏感词后触发审核，维护评论树结构。

deleteComment(commentId, operatorId)：软删除评论，支持用户自主删除或管理员操作。

likeComment(userId, commentId)：处理评论点赞，更新点赞数。

getComments(videoId, page, sort)：获取评论列表，支持按时间或热度排序。

approveComment(commentId)：审核通过后显示评论。

协作关系：

与审核子系统集成进行内容审核（后置审核）。

与 MessageService 联动发送回复通知。

5.3.2.2 控制器类

1. ManagerUserController

职责：管理用户生命周期（注册、登录、资料编辑），对应用例“成为正式用户”（3.4.1.2）。

协调流程：

UserService.register()验证信息 → 创建账户 → 返回结果。

UserService.login()验证 → 返回令牌。

异常处理：捕获用户名已存在等，返回友好错误（如“用户名已存在”）。

2. VideoController

职责：处理视频播放、互动及信息查询，对应用例“观看视频”（3.4.1.1）和“观看并互动视频”（3.4.2.1）。

协调流程：

VideoService.getVideoInfo()获取元数据 → 调用 VideoService.playVideo()返回流。

InteractionService.likeVideo()更新数据。

3. VideoSendController

职责：专处理视频上传流程，与 VideoController 互补，对应用例“发布视频”（3.4.3.1）。

协调流程：接收上传请求 → 调用 VideoService.uploadVideo()验证文件 → 提交审核队列。

4. CommentController

职责：管理评论操作，对应用例中的评论互动（3.4.2.1）。

协调流程：接收评论请求 → 调用 CommentService.addComment()校验内容 → 触发审核。

5. PrivateMessageController

职责：处理私信发送/接收，对应用例“与其他正式用户交流”（3.4.2.2）。

协调流程：接收私信 → 调用 MessageService.sendPrivateMessage()→ 校验接收者状态。

6. FriendCircleController

职责：管理朋友圈动态发布/互动，对应用例动态分享（3.4.2.2）。

协调流程：接收动态请求 → 调用 MessageService.postFriendCircleMessage()应用可见性规则。

7. ResourceManagerController

职责：管理系统资源（如文件存储、缓存），作为与基础设施层的桥梁。

协调流程：处理存储分配 → 调用外部云存储 API。

8. PerformanceAnalysisController

职责：监控系统性能，对应用例“分析系统性能”（3.4.5.1）。

协调流程：定时收集指标 → 生成报告 → 触发警报（如响应时间超标）。

9. AlertManagementController

职责：处理系统警报，对应用例“更新系统”（3.4.5.2）。

协调流程：接收警报 → 调用 PerformanceAnalysisController 定位问题 → 通知管理员。

5.3.3 领域层

领域层是系统的业务核心，负责封装与技术和平台无关的业务逻辑与规则。其目标是实现与外部框架、持久化和用户交互的解耦，确保业务逻辑的独立性和可测试性，从而灵活应对未来业务规则的变化。

职责：封装操作，确保业务逻辑的独立性

5.3.3.1 实体：User

说明：作为系统中最核心的实体之一，代表了一个唯一的参与者。

设计决策：在健壮性分析阶段我们将 user 分为多个类，如 auditor,verifiedUser,visitor,现在我们使用一个类 User,User 通过组合关系，可以同时有一个或多种身份来对应上述描述的类。

ContentCreator（up 主），Auditor，VerifiedUser，Video，Comment

值对象： userId（唯一标识），userName，identity（角色标识）

领域服务（Domain Services）：用户登陆认证，推荐算法，搜索，消息通知

仓储接口（Repository Interface）：用户，视频，弹幕，评论，审核仓储接口

5.3.3.2 实体：Video

说明：代表平台上的一个视频内容单元，是业务的核心资产。

值对象：videoId（唯一标识）、title（标题）、description（简介）、status（状态，如审核中/已发布）、fileInfo（文件信息）、updataTime（更新时间）、authorId（作者 ID）

子实体：Comment

行为：1.publish（ ）执行视频发布的核心领域逻辑，包括状态校验、初始化互动数据等。

2.play（ ）执行播放视频的逻辑操作，包括获取视频信息等。

3.addComment（ ）执行添加评论的信号发送，发送给 Comment 进行具体的业务逻辑。

4.addView(),addLike(), addCoin(), addForward()：处理相应互动操作的业务逻辑。

5.approve()处理视频审核通过状态变迁及相关逻辑。

6.reject( ) 处理视频审核驳回状态变迁及相关逻辑。

5.3.3.3 实体：Comment

说明：代表对视频的用户评论，一个视频中有多条评论。

值对象：commentId（评论唯一 id）、commentContent（评论内容）、commentLike（该评论点赞数）、commentParent（父评论 id）、videoId（视频 id）、status（评论状态）

父实体：Viedo

行为：1.addComment（ ）处理评论的创建，包含内容合规性初筛。

2.deleteComment（ ）处理评论的删除。

3.likeComment（ ）处理评论点赞的逻辑。

4.reply（ ）处理回复评论的逻辑。

5.3.3.4 领域服务

说明：不属于任何一个实体，其操作是封装的，无状态的。

UserAuthenticationService（用户认证服务）：封装复杂的用户登录认证逻辑，涉及密码验证、第三方令牌校验等。

RecommendationService（推荐服务）：封装基于用户行为、视频内容、社交关系等生成个性化视频流的核心推荐算法。此逻辑涉及多个实体。

SearchService（搜索服务）：封装复杂的搜索逻辑，如分词、权重计算等。

5.3.4 数据访问层

职责：负责与数据进行通信，数据库操作，对象关系映射，数据映射，数据持久化

5.3.4.1 数据通信

1.将对象的数据放到数据库表中

2.将数据库中的数据在对象之间传输使用，并同步更新数据到内存

5.3.4.2 数据操作封装

1.简单封装增删查改操作。

2.封装复杂的查询，如弹幕列表，评论列表等

5.3.4.3 对象关系映射

1.将内存中的对象转换成表中对应的数据

2.映射对象之间的关系到数据库表关联

3.例子：将正式用户的一个具体对象转换成数据库正式用户表中的一个具体的表项;关注关系（多对多），用户发布视频的关系（一对多）

5.3.4.4 数据映射

1.将数据库表查询结果转换成具体对象。

2.例子：正式用户的一个表项转换成一个具体的正式用户对象

5.3.4.5 数据一致性

1.并发控制，对数据冲突可采用饥饿锁

2.采用事务管理确保数据操作的原子性、一致性。

3.例子：用户投币时，确保硬币数量在并发情况下准确扣减；视频播放数，历史记录，三连数的并发更新;

5.4. 子系统划分与职责

社交互动子系统

核心职责：负责平台内所有用户间的互动行为，是构建社区氛围与用户粘性的关键。

关键功能：评论的发布与管理、回复、点赞/点踩；用户间的关注与粉丝关系维护；私信通信；动态的发布与展示。

管理核心：评论、回复、用户关系、私信、动态等互动数据。

观看视频子系统

核心职责：为用户提供稳定、流畅、沉浸式的视频消费体验，是平台最核心的用户侧功能。

关键功能：视频流的加载与播放、清晰度切换、播放进度控制、弹幕的实时渲染与显示、播放历史记录。

关联质量属性：直接关联“视频播放点击后开始缓冲时间<5 秒”的性能需求。

发布与管理视频子系统

核心职责：为 UP 主提供高效、便捷的视频内容创作与生命周期管理工具，是平台内容生态的供给端。

关键功能：视频文件与元数据（标题、封面等）的上传；已发布视频的编辑、更新、删除、隐藏；视频数据（播放量、互动数）的统计与展示。

协作关系：视频上传后，会向审核子系统提交审核请求。

消息通知子系统

核心职责：作为系统的信息神经中枢，负责系统内各类事件触发的用户通知的生成与分发。

关键功能：系统公告的推送；用户相关的互动通知（如被点赞、被回复、新粉丝）；UP 主相关的流程通知（如视频审核结果）；实时聊天的信令传递。

审核子系统

核心职责：保障平台内容安全与合规，维护健康的社区环境。

关键功能：对用户生成的视频、弹幕、评论、动态等内容进行审核（自动规则与人工审核结合）；对违规用户进行封禁等处理；提供审核员工作台。

关联质量属性：支撑“人工审核+关键词自动检测”的合规性需求。

系统管理子系统

核心职责：为系统管理员提供全局性的监控、配置与管理能力，确保平台的稳定、安全运行。

关键功能：全局用户管理与权限配置；系统关键参数的设置与调整；监控系统健康状况（服务器资源、服务状态）并生成数据报表；系统更新与维护。

搜索与推荐子系统

核心职责：连接用户与内容，提升内容分发效率与用户内容发现体验。

关键功能：

搜索：基于关键词的多条件视频搜索，支持相关性排序。

推荐：基于用户行为、兴趣标签、社交关系等，生成个性化的首页视频推荐流。

关联质量属性：直接关联“首页首屏加载时间<3 秒”的性能需求。

用户管理子系统子系统

核心职责：管理平台所有用户的身份、认证与基础信息，是系统安全的基础。

关键功能：用户注册、登录、登出、第三方授权；个人基本信息（昵称、签名）的管理；密码修改与安全设置。

6 详细设计

6.1.

6.2.

6.3. 用况转化输入输出

6.3.1 与其他正式用户交流用况

6.3.1.1 UI 层-PrivateMessageUI

类名	PrivateMessageUI
功能定位	**前端界面交互入口**，负责私信界面的展示、私信发送触发及启动流程控制
关联方法	+showPrivateMessage() +sendPrivateMessage() +startInterface()
输入来源	
输入内容	
约束条件	**无数据格式校验**（校验由 PrivateMessageController 处理）
输出目标	
输出内容	Bool + String（发送成功/失败 + 提示语，如 *“私信发送成功”*）
输出场景	
处理步骤关联	startInterface()→ 渲染私信界面 2. 触发 sendPrivateMessage()→ 传递“发送”操作事件至 PrivateMessageController

6.3.1.2 UI 层 - FriendCircleUI

类名	FriendCircleUI
功能定位	**前端界面交互入口**，负责朋友圈界面展示、发布/点赞触发及启动流程控制
关联方法	+showFriendCircle() +publishFriendCircle() +startInterface() +likeFriendCircle()
输入来源	
输入内容	
约束条件	**无数据格式校验**（校验由 FriendCircleController 处理）
输出目标	
输出内容	Bool + String（操作成功/失败 + 提示语，如 *“朋友圈发布成功”*）
输出场景	
处理步骤关联	startInterface()→ 渲染朋友圈界面 2. 触发 publishFriendCircle()→ 传递“发布内容”至 FriendCircleController 3. 触发 likeFriendCircle()→ 传递“点赞目标”至 FriendCircleController

6.3.1.3 Controller 层 - PrivateMessageController

类名	PrivateMessageController
功能定位	**业务逻辑核心**，负责收发流程控制、消息存储与历史查询

关联属性	− userInbox : unordered_map<string, vector<PrivateMessage*>> − userOutbox : unordered_map<string, vector<PrivateMessage*>> − messageStore : unordered_map<string, PrivateMessage>
关联方法	+getPrivateMessageHistory(senderId:string, receiverId:string) : PrivateMessage[] +sendPrivateMessage(content:string, senderId:string, receiverId:string) : boolean
输入来源	PrivateMessageUI）的操作请求、系统内部消息触发（如定时任务，若需扩展）
输入内容	senderId/receiverId 字符串、content 字符串；“发送请求事件”“查询请求事件”）
约束条件	senderId/receiverId 需为系统中已存在的 VerifiedUser.userId 2. content 非空（长度规则由 UI 层前置校验或 Controller 二次校验） 3. 消息存储需保证原子性（如多线程场景下的锁机制）
输出目标	PrivateMessageUI）的结果反馈、数据存储层的消息持久化（需关联数据库操作类，图中未显式体现则默认内部处理）
输出内容	Bool + String（操作成功/失败 + 提示语，如 *“私信发送成功”*） PrivateMessage[]（私信历史列表，用于界面渲染）
输出场景	
处理步骤关联	sendPrivateMessage 请求 → 解析参数 → 校验合法性 → 创建 PrivateMessage 实例 → 写入 userOutbox/sender+ userInbox/receiver→ 持久化至 messageStore→ 反馈 UI 层结果 2. 接收 getPrivateMessageHistory 请求 → 解析 senderId/receiverId→ 从 userInbox/receiver 查询消息 → 封装为数组 → 反馈 UI 层

6.3.1.4 Controller 层 - FriendCircleController

类名	FriendCircleController
功能定位	**业务逻辑核心**，负责发布流程控制、历史查询及审核关联
关联属性	− userCircles : unordered_map<string, vector<FriendCircle*>> − circleStore : unordered_map<string, FriendCircle>
关联方法	+getAllFriendCircleHistory(userId:string) : vector<FriendCircle> +publishFriendCircle(content:string, publisherId:string) : boolean
输入来源	FriendCircleUI）的操作请求、审核模块（Auditor 相关逻辑，若耦合）
输入内容	userId 字符串、content 字符串；“发布请求事件”“查询请求事件”）
约束条件	publisherId 需为系统中已存在的 VerifiedUser.userId 2. content 非空（长度规则由 UI 层前置校验或 Controller 二次校验） 3. 发布时需触发审核流程（调用 FriendCircle.checkFriendCircle()或关联 Auditor 逻辑）
输出目标	FriendCircleUI）的结果反馈、数据存储层的朋友圈持久化（需关联数据库操作类，图中未显式体现则默认内部处理）
输出内容	Bool + String（操作成功/失败 + 提示语，如 *“朋友圈发布成功”*） vector<FriendCircle>（朋友圈历史列表，用于界面渲染）
输出场景	
处理步骤关联	publishFriendCircle 请求 → 解析参数 → 校验合法性 → 创建 FriendCircle 实例 → 调用 FriendCircle.checkFriendCircle()审核 → 写入 userCircles/publisher+ circleStore→ 然后加进数据库持久化 → 反馈 UI 层结果 2. 接收 getAllFriendCircleHistory 请求 → 解析 userId→ 从 userCircles/user 查询朋友

圈 → 封装为列表 → 反馈 UI 层

6.3.1.5 Model 层 - FriendCircle（实体类）

类名	FriendCircle
功能定位	**实体模型**，封装帖子核心信息及基础操作方法
属性	− postId : string − content : string − postTime : Date − isApproved : boolean
方法	+checkFriendCircle(circleId:string) : boolean +getAllFriendCircleHistory() : vector<FriendCircle>
关联关系	VerifiedUser（发布者/浏览者）、Auditor（审核者）、FriendCircleController 交互
输入来源	Controller 层（FriendCircleController）的操作请求、系统时间（postTime 自动生成）
输入内容	circleId 字符串、content 字符串；“发布请求事件”“查询请求事件”）
约束条件	postId 需全局唯一（生成规则如 UUID） 2. postTime 为操作时的系统时间（精度到秒） 3. isApproved 初始值为 false，审核通过后置为 true
输出目标	Controller 层（反馈审核结果、历史查询结果）、UI 层（反馈朋友圈列表数据）
输出内容	boolean（审核是否通过） vector<FriendCircle>（朋友圈历史列表）
输出场景	
处理步骤关联	Controller 创建实例 → 调用 checkFriendCircle()⟶ 返回审核结果给 Controller 2. 查询历史时，Controller 调用 getAllFriendCircleHistory()⟶ 返回朋友圈列表给 Controller

6.3.1.6 Model 层 - Auditor（实体类）

类名	Auditor
功能定位	**实体模型**，标识审核人员身份，关联朋友圈审核流程
属性	− auditorId : string − auditorName : string
方法	+getAuditor() : Auditor
关联关系	FriendCircle（审核对象）、FriendCircleController（审核流程触发）交互
输入来源	Controller 层调用）
输入内容	auditorId 字符串、auditorName 字符串；“审核请求事件”）
约束条件	auditorId 需全局唯一 2. auditorName 非空
输出目标	Controller 层（反馈审核员信息）、系统日志（可选，记录审核操作）
输出内容	Auditor 实例（包含审核员身份信息）
输出场景	
处理步骤关联	Auditor 实例并存入系统 2. 朋友圈审核时，Controller 调用关联逻辑 → 记录审核人信息

6.3.1.7 Model 层 - VerifiedUser（实体类）

类名	VerifiedUser
功能定	**实体模型**，封装用户核心信息及社交功能操作方法（私信、朋友圈）

位	
属性	− userName : string − userId : string
方法	+getUser(userId:string) : VerifiedUser +sendPrivateMessage(content:string, senderId:string, receiver:string) : boolean +publishFriendCircle(content:string, publisherId:string) : boolean
关联关系	PrivateMessage（私信发送者/接收者）、FriendCircle（朋友圈发布者）、PrivateMessageController（私信操作）、FriendCircleController（朋友圈操作）交互
输入来源	PrivateMessageUI、FriendCircleUI）的用户操作、系统用户管理模块
输入内容	userId 字符串、content 字符串、receiver 信息；“发送/发布请求事件”）
约束条件	userId 需全局唯一（注册时生成） 2. userName 非空
输出目标	Controller 层（反馈操作结果）、UI 层（反馈社交功能结果）
输出内容	boolean（操作是否成功） 结构化数据（如私信发送状态、朋友圈发布状态）
输出场景	
处理步骤关联	Controller 调用 sendPrivateMessage()→ 操作成功后反馈 UI 2. 发布朋友圈时，UI 层触发请求 → Controller 调用 publishFriendCircle()→ 审核通过后发布并反馈 UI

6.3.1.8 Model 层 - PrivateMessage（实体类）

类名	PrivateMessage
功能定位	**实体模型**，封装私信核心信息及基础操作方法
属性	− messageId : string − content : string − sendTime : Date
方法	+savePrivateMessageHistory() : void +getPrivateMessageId() : string
关联关系	VerifiedUser（发送者/接收者）、PrivateMessageController（私信存储）、PrivateMessageUI（私信展示）交互
输入来源	Controller 层（PrivateMessageController）的操作请求、系统时间（sendTime 自动生成）
输入内容	messageId 字符串、content 字符串；“存储请求事件”）
约束条件	messageId 需全局唯一（生成规则如 UUID+时间戳） 2. sendTime 为发送时的系统时间（精度到毫秒）
输出目标	Controller 层（反馈存储结果）、UI 层（反馈私信内容）
输出内容	void（存储完成无返回，或布尔值表示成功） messageId 字符串（用于查询唯一私信）
输出场景	
处理步骤关联	Controller 创建实例 → 调用 savePrivateMessageHistory()→ 持久化至 messageStore 然后加进数据库→ 反馈 UI 层

、

6.3.1.9 与其他正式交流数据库表

1.朋友圈表

字段名	数据类型	约束条件	备注
post_id	VARCHAR(64)	PRIMARY KEY, UNIQUE	UUID 生成
publisher_id	VARCHAR(64)	FK(users.user_id)	发布者 ID
content	TEXT	NOT NULL	动态内容
media_urls	JSON		附件 URL 列表（图片/视频）
visibility	ENUM	DEFAULT 'friends'	可见性 ('public','friends','private')
approved_at	TIMESTAMP		审核通过时间
is_deleted	BOOLEAN	DEFAULT FALSE	逻辑删除标记

2.私信表

字段名	数据类型	约束条件	备注
message_id	VARCHAR(64)	PRIMARY KEY, UNIQUE	UUID 生成
sender_id	VARCHAR(64)	FK(users.user_id)	发送者 ID
receiver_id	VARCHAR(64)	FK(users.user_id)	接收者 ID
content	TEXT	NOT NULL	私信内容
read_status	BOOLEAN	DEFAULT FALSE	是否已读
sent_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP	发送时间
is_deleted	BOOLEAN	DEFAULT FALSE	逻辑删除标记

后记

参考文献

[1] YOUNG.RSS 是什么？[EB/OL]. http://jingpin.org/what-is-rss/.

[2] 杨博，彭博.RSS 提要分析与阅读器设计[R].成都：四川大学计算机学院，2007：42-43.

[3] 逸出络然.RSS 技术的原理[EB/OL].http://yclran.blog.163.com/blog/static/9794549620091110341115 58/.

[4] 佚名.Qt 是什么[EB/OL]. http://qt.nokia.com/title-cn.

[5] 佚名.Model/View Programming[EB/OL]. http://doc.trolltech.com/4.6/model-view-programming.html.

[6] [加拿大]Jasmin Blanchette[英]Mark Summerfield 著 闫锋欣,曾泉人,张志强译.

[7] C++ GUI Qt4 编程（第二版）[M].电子工业出版社：2008:182-206,291-305.

[8] 佚名.XML Processing[EB/OL]. http://doc.trolltech.com/4.6/xml-processing.html.

[9] Michael Blala James Rumbangh 著.UML 面向对象建模与设计（第 2 版）[M].北京：人民邮电出版社,2006:136-235.

[10] 胡海静,王育平,等. XML 技术精粹[M]. 北京：机械工业出版社，2001:17-19.